

Programme du 7 mai 2026 d'examen et
d'évaluation des candidatures à l'admission en
première année du parcours *Informatique & droit*
pour l'année universitaire 2026 → 2027

La commission d'examen des candidatures au parcours Informatique et Droit
du Master Droit du numérique de l'université Paris est Créteil,

Vu le règlement UE 2016/679 du Parlement européen et du Conseil du 27 avril
2016 relatif à la protection des personnes physiques à l'égard du traitement
des données à caractère personnel et à la libre circulation de ces données, et
abrogeant la directive 95/46/CE,

Vu le code de l'éducation, notamment ses articles L. 612-5, L. 612-6, L. 613-1,
D. 123-13, D. 336-3, D. 612-36-2 à D. 612-36-2-12, D. 613-3, D. 613-6, D. 711-1 et
R. 732-1,

Vu le code du patrimoine, notamment son livre II,

Vu le code des relations entre le public et l'administration, notamment ses
articles L. 300-4, L. 311-1, L. 311-3-1 à R. 311-3-1-2 et L. 312-1-3,

Vu la loi n° 78-17 du 6 janvier 1978 relative à l'informatique, aux fichiers et aux
libertés,

MODIFIÉE



Vu l'arrêté du 25 avril 2002 relatif au diplôme national de master,

MODIFIÉ

Vu l'arrêté du 22 janvier 2014 fixant le cadre national des formations conduisant
à la délivrance des diplômes nationaux de licence, de licence professionnelle et
de master,

MODIFIÉ

Vu l'arrêté du 9 février 2023 portant création d'un traitement automatisé de
données à caractère personnel dénommé « Mon Master »,

MODIFIÉ

Vu l'arrêté du 13 février 2026 relatif au calendrier de la procédure dématérialisée
de candidature et de recrutement en première année des formations conduisant
au diplôme national de master au titre de l'année universitaire 2026-2027,



ABROGÉ, mais en tant qu'il a produit des effets juridiques pris en compte dans l'évaluation des candidatures

MODIFIÉ

MODIFIÉ

Vu l'arrêté du 17 janvier 1992 relatif aux voies d'orientation,

Vu l'arrêté du 16 juillet 2018 relatif à l'organisation et aux volumes horaires des enseignements du cycle terminal des lycées, sanctionnés par le baccalauréat général,

Vu l'arrêté du 16 juillet 2018 portant organisation et volumes horaires des enseignements des classes de première et terminale des lycées sanctionnés par le baccalauréat technologique, séries « sciences et technologies de la santé et du social (ST2S) », « sciences et technologies de laboratoire (STL) », « sciences et technologies du design et des arts appliqués (STD2A) », « sciences et technologies de l'industrie et du développement durable (STI2D) », « sciences et technologies du management et de la gestion (STMG) », « sciences et technologies de l'hôtellerie et de la restauration (STHR) »,

Vu l'arrêté du 31 juillet 2019 fixant le programme des enseignements de spécialité des classes de première et terminale conduisant au baccalauréat technologique série sciences et techniques du théâtre, de la musique et de la danse (S2TMD),

Vu l'arrêté du 28 juillet 2025 relatif aux épreuves du baccalauréat technologique série « sciences et technologies de l'agronomie et du vivant » (STAV) préparé dans les établissements de l'enseignement agricole,

ABROGÉ, mais en tant qu'il a produit des effets juridiques pris en compte dans l'évaluation des candidatures

Vu l'arrêté du 1^{er} mars 2016 portant création de la spécialité « Systèmes numériques » de baccalauréat professionnel comportant trois options et fixant ses modalités de délivrance,

Vu l'arrêté du 13 février 2023 portant création de la spécialité « cybersécurité, informatique et réseaux, électronique » de baccalauréat professionnel et fixant ses modalités de délivrance,

ABROGÉ, mais en tant qu'il a produit des effets juridiques pris en compte dans l'évaluation des candidatures

Vu l'arrêté du 1^{er} mars 2016 portant création de la spécialité « Métiers de l'électricité et de ses environnements connectés » de baccalauréat professionnel et fixant ses modalités de délivrance,

Vu l'arrêté du 8 janvier 2024 portant création de la spécialité « Métiers de l'électricité et de ses environnements connectés » de baccalauréat professionnel et fixant ses modalités de délivrance,

ABROGÉ, mais en tant qu'il a produit des effets juridiques pris en compte dans l'évaluation des candidatures

Vu l'arrêté du 30 mai 2005 portant création du baccalauréat professionnel spécialité « maintenance des équipements industriels » et fixant ses modalités de préparation et de délivrance,

Vu l'arrêté du 9 décembre 2020 portant création de la spécialité « Maintenance des systèmes de production connectés » de baccalauréat professionnel et fixant ses modalités de délivrance,

Vu l'arrêté du 25 septembre 2021 relatif au certificat de capacité en droit,



Vu l'arrêté (MESRI – DGESIP BI-1) du 19 mars 2020 portant accréditation de l'université Paris-XII en vue de la délivrance de diplômes nationaux,



VU la délibération CFVU-2025-FORMATION-63 de la commission de la formation et de la vie universitaire de l'université Paris-XII du 12 décembre 2025 approuvant la campagne nationale Mon Master 2026 - examen des données obligatoires (capacités d'accueil, modalités de recrutement, critères d'examen des candidatures (CGEC), attendus et mode de composition des commissions de recrutement,

https://www.u-pec.fr/medias/fichier/63-del-cfvu-campagne-nationale-mon-master-2026_1776948310463-pdf

VU la délibération CA-2025-FORMATION-93 du conseil d'administration de l'université Paris-XII du vendredi 19 décembre 2025 portant approbation des données obligatoires des Masters 1^{ère} année de « monmaster.gouv.fr » - campagne 2026 : capacités d'accueil, modalités de recrutement, critères d'examen des candidatures (CGEC), attendus et mode de composition des commissions de recrutement,

https://www.u-pec.fr/medias/fichier/del-ca-approbation-des-donnees-obligatoires-relatives-a-la-campagne-master-2026_1768918310182-pdf



VU le système de composition T_EX version 3.141592653 du 9 février 2021, notamment le moteur luaH_BT_EX, le format L^AT_EX, la classe memoir et les paquets pgf/tikz et pdfpages,

<https://svn.tug.org:8369/texlive/?revision=78226>

VU l'éditeur emacs version 30.2 du 14 août 2025,

<https://cgit.git.savannah.gnu.org/cgit/emacs.git/tag/?h=emacs-30.2>

VU le langage C_at_al_a version 1.1.0 « bac d'Eloka » du 26 janvier 2026,

<https://github.com/CatalaLang/catala/releases/tag/1.1.0>

VU le langage OC_am_l version 5.4.1 du 17 février 2026, notamment les bibliothèques Re et Csv et le moteur de production dune,

<https://github.com/ocaml/ocaml/releases/tag/5.4.1>

VU l'interpréteur fish version 4.6.0 du 28 mars 2026,

<https://github.com/fish-shell/fish-shell/releases/tag/4.6.0>



VU le contrat de licence de logiciel libre CeCILL version 2.1 du 21 juin 2013,

https://cecill.info/licences/Licence_CeCILL_V2.1-fr.txt

VU l'analyse d'impact relative à la protection des données du présent programme version 2.0 du 6 mai 2026,



CONSIDÉRANT qu'il appartient à la commission d'examen des candidatures d'apprécier les candidatures au regard des attendus et des critères généraux d'examen de la formation concernée, et d'en organiser la mise en oeuvre,

CONSIDÉRANT que l'usage maîtrisé et proportionné de la programmation informatique peut rendre les procédures administratives plus justes et contestables,

CONSIDÉRANT qu'il y a lieu d'assurer l'égalité de traitement des candidatures, en fondant leur évaluation sur des critères objectifs et en ne prévoyant de différences de traitement qu'entre des situations distinctes, en rapport avec l'objet de l'évaluation et proportionnées à ces différences,

CONSIDÉRANT que les modalités d'évaluation doivent permettre de rendre effectif le droit au recours contre les décisions prises, en assurant la traçabilité des opérations, l'explicitation des critères appliqués et la possibilité de reconstituer le processus ayant conduit à chaque décision,

<https://www.vie-publique.fr/files/rapport/pdf/302817.pdf>

Raphaël Monat et Liane Huttner (2026). « Coding computational laws: 20 recommendations for public administrations ». In : *Information and Communications Technology Law* 35 (2)

CONSIDÉRANT les recommandations issues du 8^e rapport annuel au Parlement du Comité éthique et scientifique de Parcoursup et MonMaster,

CONSIDÉRANT les recommandations de Raphaël Monat et Liane Huttner pour la programmation d'un droit calculatoire,

P R O G R A M M E

I Principes

- 1 Le présent programme comprend l'ensemble des opérations automatisées et non automatisées permettant l'examen et l'évaluation des candidatures au Master Droit du numérique – parcours Informatique et Droit de l'université Paris XII (université Paris-Est Créteil, ci-après « UPEC »), en vue de leur classement par la commission d'examen des candidatures (ci-après « commission »).
- 2 Les candidatures concernées par ce programme sont celles qui sont fournies à la commission par la plateforme MonMaster, dans le cadre de la procédure dématérialisée de recrutement et de préparation à l'inscription en première année des formations conduisant au diplôme national de master (traitement de données à caractère personnel dénommé « Mon Master »).
- 3 La procédure d'examen et d'évaluation des candidatures en vue de leur classement a lieu exclusivement sur dossier; il n'y a ni entretien ni épreuve d'admission. Le dossier de chaque candidat·e se décompose en deux parties :
 - des réponses à un formulaire, ci-après « formulaire », dans un format décrit à l'Annexe C;
 - des pièces-jointes, ci-après « pièces ».

Aucune nouvelle information ou pièce n'est demandée aux candidat·es. Toutefois, la possibilité est donnée aux candidat·es de rectifier l'interprétation faite par le présent programme des informations saisies dans la plateforme MonMaster (voir Section 3.1).

- 4 Chaque élément produit dans le cadre du programme est caractérisé à la fois par son statut et son mode de production. Son statut précise sa fonction dans le processus: le programme engendre des documents de travail — produits par les membres de la commission en suivant les instructions du programme ou des fonctions ci-après décrites — ainsi que des sorties. Son mode de production indique s'il résulte d'une intervention d'un·e, de plusieurs ou de l'ensemble des membres de la commission appliquant le programme ou de l'exécution automatisée d'une de ses étapes.
- 5 L'attribution, par la commission, d'un rang de classement ou d'un motif de refus à chaque candidature, constitue la sortie principale du programme — selon un format spécifié par la plateforme MonMaster, voir § 155. Cette sortie se matérialise par une liste unique de candidatures, indiquant, pour chacune

d'elles, son rang de classement parmi les candidatures admises ou le motif de refus qui lui est opposé.

- 6 Une deuxième sortie est constituée des notices individuelles d'information, transmises à chaque candidat·e après qu'ils aient reçu, via la plateforme MonMaster, le résultat de l'examen de leurs candidatures. Cette notice individuelle d'information donne à chaque candidat·e son score si le programme lui en a attribué un et, s'il a fait l'objet d'un rang de classement, son rang; elle lui fournit, en outre, un lien pour accéder au présent programme, ainsi qu'une notice explicative de la manière dont ce dernier a été mis en œuvre.
- 7 Une troisième sortie est constituée d'éléments statistiques, sur la base desquels le rapport de la commission sera élaboré par la commission, puis publié par les directeur·ices du master, notamment via la plateforme MonMaster conformément au 3^{ème} alinéa de l'article D. 612-36-2-1 du code de l'éducation.
- 8 La commission nomme en son sein un·e opérateur·ice du traitement. L'opérateur·ice du traitement est chargé·e, sous le contrôle des autres membres de la commission, de récupérer les dossiers sur MonMaster, exécuter les fragments informatiques du programme, récupérer les documents de travail validés par la commission et communiquer aux rapporteur·ices les documents de travail produits par les fragments informatiques du programme.
- 9 Le président·e de la commission est chargé·e de téléverser la sortie principale du programme sur la plateforme MonMaster. Ce téléversement constitue la modalité retenue pour la mise en oeuvre, à l'UPEC, des obligations prévues au premier alinéa de l'article D. 612-36-2-2-1 du code de l'éducation.
- 10 Chaque fois que de besoin, la commission nomme en son sein des rapporteur·ices, chargé·es de l'exécution des fragments humains du programme. Sur la base de documents de travail fournis par l'opérateur·ice, les rapporteur·ices préparent des documents de travail, sur lesquels la commission s'appuie pour délibérer.
- 11 Aucune décision relative à une candidature n'est intégralement automatisée : chaque candidature fait l'objet d'un examen individuel effectif par un·e rapporteur·ice, chargé·e de la préparation de documents de travail la concernant, lesquels sont soumis à la validation collégiale de la commission. Le programme ne constitue pas pour autant un simple outil d'aide à la décision : il est le cadre opératoire au sein duquel la commission, en tant que jury souverain chargé de l'établissement du classement des candidatures, exerce sa compétence, en organisant un enchaînement d'opérations, automatisées et non automatisées, dont elle assure la conception, le paramétrage et la mise en œuvre.
- 12 Les fragments exécutés sur ordinateur du programme ne se basent que sur les informations du formulaire et sur les documents de travail validés par la commission. Ils sont écrits dans les langages de programmation `Catala`, `OCaml`, `emacs-lisp`, `fish` et `TEX`.

Donald Everett Knuth (1984).
« Literate Programming ». In:
The Computer Journal 27 (2)

- 13 Le programme est écrit selon les principes de la programmation lettrée. Il est divisé en paragraphes numérotés. Chaque paragraphe consiste en de la prose et/ou des fragments formels. Les fragments formels sont nommés par un nom en PETITES CAPITALES SANS EMPÂTEMENTS entre chevrons $\langle \dots \rangle$.

Quand un tel nom apparaît suivi du symbole \equiv , cela signifie que le nom doit être défini par le fragment qui le suit.

Quand un tel nom, suivi d'un numéro de paragraphe, apparaît suivi du symbole $+ \equiv$, cela signifie qu'à sa définition qui commence au paragraphe numéroté doit être ajoutée le fragment qui suit, à la suite de sa définition précédente.

Quand un tel nom apparaît dans un fragment formel, il est suivi du numéro de paragraphe où il est défini. Cela signifie que que le nom doit être remplacé textuellement par sa définition intégrale, précédé le cas échéant par le début de la ligne où il est présent, de manière récursive.

- 14 Quand un nom est remplacé par le nom d'un fichier (écrit en police à chasse fixe), alors sa définition est écrite dans ledit fichier, permettant l'exécution du programme avec les outils usuels d'exécution.

- 15 Les fragments en OCaml sont légèrement modifiés pour l'affichage, suivant la table ci-dessous :

source	affichage
<code>-></code>	\rightarrow
<code><-</code>	\leftarrow
<code>>=</code>	\geq
<code>*</code>	\times
<code>fun</code>	λ

- 16 Le présent document constitue le programme contenant l'intégralité de l'information nécessaire à son exécution. En cela, ce programme est un code source.

<https://lacl.fr/~lpellissier/master/programme-monmaster-2026.org>

- 17 En plus de sa version PDF, ce programme est aussi disponible sur la page web du master dans son format usuel de modification en org-mode. En cela, il est mis à disposition dans un standard ouvert, aisément réutilisable et exploitable par un système de traitement automatisé, conformément aux dispositions de l'article L. 300-4 du code des relations entre le public et l'administration.

- 18 Sous ce format, il peut être tissé (*weaved*) en le présent document ou enchevêtré (*tangled*) en les fichiers soutenant l'exécution du programme. L'intégralité des fichiers enchevêtrés peuvent aussi être produit manuellement à partir du présent document, en version org-mode comme en version PDF.

- 19 Le programme est conçu de manière à répondre par lui-même à un niveau élevé d'exigences en matière de redevabilité des traitements algorithmiques, y compris au-delà des obligations légales applicables.

À ce titre, il permet :

- a) de donner accès à toute personne aux règles définissant le traitement ainsi qu'aux caractéristiques de sa mise en œuvre, sous une forme intelligible, et en premier lieu aux personnes faisant l'objet des décisions individuelles

prises sur le fondement dudit traitement conformément aux dispositions des articles L. 311-3-1 à R. 311-3-1-2 du code des relations entre le public et l'administration ;

- b) de fournir à toute personne — et en premier lieu aux personnes concernées —, des informations utiles concernant la logique sous-jacente du traitement, indépendamment de l'applicabilité de l'article 22 du règlement UE 2016/679 du 27 avril 2016 susvisé ;
- c) d'assurer la capacité du responsable de traitement à maîtriser le fonctionnement et les évolutions du traitement, et à en expliquer, de manière détaillée et intelligible, la mise en œuvre à l'égard des personnes concernées, indépendamment de l'applicabilité de l'article 47 de la loi n° 78-17 du 6 janvier 1978 susvisée.

2 Architecture du programme

- 20 Au regard des attendus de la formation, et notamment des attendus 1 (« *Disposer de solides connaissances en droit (attestées par de très bons résultats en premier cycle universitaire)* ») et 2 (« *Avoir une disposition positive envers les mathématiques et les formalismes mathématiques: les candidates et candidats devront avoir témoigné d'un intérêt (ou au moins un non-rejet) pour les mathématiques lors de leur formation au lycée* »), et de la qualité variable du remplissage du formulaire, les candidatures sont ainsi réparties en plusieurs groupes permettant de les comparer de manière homogène.

Ces groupes sont définis sur des critères objectifs, liés aux caractéristiques du baccalauréat, au parcours universitaire et aux résultats académiques, ainsi qu'à l'impossibilité d'effectuer les calculs de manière automatisée du fait de la mauvaise qualité du remplissage du formulaire.

`<TYPESDOSSIERS> ≡ OCaml`

```
1 type groupes_formulaire =
2   <GROUPESFORMULAIRE 28>
```

Étendu en §23.

- 21 La séparation des candidatures en groupes se fait de manière totalement automatisée. Elle repose sur une première étape de construction des données d'entrée. Une fonction du programme lit le fichier des formulaires exporté de la plateforme MonMaster et produit, pour chaque candidature, une représentation structurée des informations traitées extraites du formulaire.

`<FONCTIONS EXPOSÉES LECTURE ÉCRITURE> ≡ OCaml`

```
1 val construction_candidats : string →
2   (string × dossier_formulaire) list
```

Étendu en §33, 162.

- 22 Cette représentation prend la forme d'une liste d'association dont la clef est le numéro de candidat (une chaîne de caractère) et la valeur une représentation du dossier du candidat, de type `dossier_formulaire`. La fonction `construction_candidats` prend en entrée le nom du fichier à lire. Cette fonction dépend fortement du format des données fournies par la plateforme

MonMaster, format non documenté et en aucun cas garanti de rester constant d'une année sur l'autre. Elle est donc fragile, et est décrite Annexe c.

- 23 Les informations effectivement utilisées par cette fonction de construction sont limitées :
- des informations d'identification et de contact (des chaînes de caractères),
 - des caractéristiques sur le baccalauréat,
 - une description des trois années universitaires (bac+1, bac+2, bac+3) permettant l'inscription,
 - et les informations (booléennes) indiquant si le/la candidat·e a entamé une année bac+4 ou bac+5, a obtenu une autre licence, et a préparé sa licence de droit à l'université Paris est Créteil.

```

<TYPESDOSSIERS 20> +≡ OCaml
1 <STRUCTURES 241>
2 type dossier_formulaire =
3 {
4   nom: string;
5   prenom: string;
6   adresse_email: string;
7   baccalaureat: CaracteristiquesBac.t;
8   bac_plus_1: annee_universitaire;
9   bac_plus_2: annee_universitaire;
10  bac_plus_3: annee_universitaire;
11  bac_plus_4: bool;
12  bac_plus_5: bool;
13  autre_licence: bool;
14  upec: bool;
15 }

```

- 24 Une seconde fonction du programme associe à chaque dossier son groupe, sur la base des informations ainsi construites. En cas de saisie défectueuse ayant entraîné une mauvaise affectation, la commission rétablit le groupe lors de l'examen.

```

<FONCTIONSÉXPOSÉESCALCUL> ≡ OCaml
1 val tri_dossier : dossier_formulaire → groupes_formulaire

```

- 25 Une fois les données construites et les candidatures réparties en groupes, le programme procède à leur traitement selon une organisation générale en trois temps: traitement des candidatures relevant du groupe nominal et du groupe adjacent, dit quasi-nominal; traitement des autres groupes; interclassement des candidatures et production des sorties.

```

<PROCÉDURE> ≡ OCaml
1 let candidats = construction_candidats
2   ( <REPERTOIREENTREES 186> ^ <FICHIERFORMULAIRE 187> )
3   <CASNOMINALETADJACENT 26>
4   <AUTRESCAS 27>
5   <INTERCLASSEMENT 154>
6   <SORTIE 155>
7   <PRODUCTIONINFORMATION 163>

```

- 26 Le traitement des candidatures relevant du groupe nominal et du groupe quasi-nominal est lui-même divisé en plusieurs phases, comprenant notamment des opérations de vérification et de rattrapage de candidatures légèrement hors du cas nominal.

`<CASNOMINALÉTADJACENT> ≡ OCaml`

- 1 `<PRODUCTIONQUASINOMINAUX 130>`
- 2 `<LECTUREQUASINOMINAUX 133>`
- 3 `<PRODUCTIONVÉRIFICATIONNOMINAUX 34>`
- 4 `<LECTUREVÉRIFICATIONNOMINAUX 44>`
- 5 `<SEUIL 59>`
- 6 `<PRODUCTIONSOUSSEUIL 58>`
- 7 `<LECTURESOUSSEUIL 65>`
- 8 `<PRODUCTIONSEUIL 68>`
- 9 `<LECTURESEUIL 70>`
- 10 `<PRODUCTIONNOMINAUX 277>`

- 27 Les autres groupes donnent lieu à des traitements spécifiques, eux-mêmes décomposés en plusieurs sous-cas (ici, quatre).

`<AUTRESCAS> ≡ OCaml`

- 1 `<PRODUCTIONAUTRELICENCEDROIT 143>`
- 2 `<LECTUREAUTRELICENCEDROIT 144>`
- 3 `<PRODUCTIONNOMINAUXSAISIEDÉFAILLANTE 138>`
- 4 `<LECTURENOMINAUXSAISIEDÉFAILLANTE 139>`
- 5 `<PRODUCTIONBUT 146>`
- 6 `<LECTUREBUT 147>`
- 7 `<PRODUCTIONAUTRE 149>`
- 8 `<LECTUREAUTRE 150>`

3 Cas nominal

- 28 Le premier groupe comprend les candidatures de personnes titulaires du baccalauréat général ou technologique, ou du certificat de capacité en droit (dit « capacité en droit »), et ayant validé les cinq ou six premiers semestres d'une licence de droit préparée dans un établissement public à caractère scientifique, culturel et professionnel au sens de l'article L. 711-1 du code de l'éducation (ci-après « université publique française »). Ce groupe est dénommé groupe *nominal*, au sens du cas de référence du programme. Voir l'Annexe D.I.

`<GROUPESFORMULAIRE> ≡ OCaml`

- 1 **Nominal**
Étendu en §128, 137, 142, 145, 148.

- 29 Le calcul applicable aux candidatures relevant du groupe nominal se déroule en plusieurs phases, dont deux relèvent d'opérations intégralement automatisées. Ces deux phases constituent le contenu d'un module `CalculScoresCasNominal`, écrit en `Catala`, traduit en `OCaml` et appelé par le reste du programme.

`<CALCULSCORESCASNOMINAL> ≡ Catala`

- 1 `<DÉPENDANCES 51>`

```

2  `` `cataLa-metadata
3  <SCOREPHASELICENCE 48>
4  <SCORECOMPLETNOMINAL 77>
5  `` `
6  <CONTENU 30>

```

- 30 Le module repose sur trois éléments : des déclarations de type, définissant les entrées et les sorties possibles des différents sous-programmes ; des coefficients fixés par la commission ; et des fonctions calculant effectivement le score, dont la logique est déterminée par la commission.

<CONTENU> ≡ Catala

```

1  `` `cataLa
2  <TYPESCASNOMINAL 31>
3  <COEFFICIENTS 55>
4  <CALCULS 80>
5  `` `

```

- 31 Les déclarations de type définissent les données nécessaires à la mise en oeuvre des opérations intégralement automatisées du traitement des candidatures du groupe nominal. Elles contiennent une modélisation des candidat·es (à chacune des deux phases), des notes de licence, et des baccalauréats (et de la capacité en droit).

<TYPESCASNOMINAL> ≡ Catala

```

1  <CANDIDATE 79>
2  <NOTESLICENCE 47>
3  <TYPESBACCALAURÉATS 82>

```

3.1 Vérification des données du formulaire

- 32 Les données du formulaire peuvent donner lieu à des erreurs d'exploitation par le programme. Ces erreurs peuvent résulter soit d'une erreur de saisie de la part des candidat·es, conduisant à une incohérence entre les données du formulaire et les pièces justificatives, soit d'une interprétation erronée des données par le programme, résultant, par exemple, d'une mauvaise compréhension par la commission du format des données du formulaire.

3.1.1 Information

- 33 Le programme contient une fonction pour écrire des fiches de vérification des informations utilisées. Cette fonction est donnée en Annexe, § 263.

<FONCTIONSÉXPOSÉESLECTUREÉCRITURE 21> + ≡ OCaml

```

1  val fiche_verification : string × dossier_formulaire → unit

```

- 34 Pour chaque candidature relevant du groupe nominal, une fiche récapitulant les informations extraites du formulaire utilisées par le programme est produite.

<PRODUCTIONVÉRIFICATIONNOMINAUX> ≡ OCaml

```

1  let () = List.iter fiche_verification candidats

```

Étendu en §38.

- 35 Cette fiche est ensuite compilée au format PDF et chiffrée. Pour chaque candidat-e, sa fiche peut être déchiffrée avec comme mot de passe le numéro de candidat-e attribué par la plateforme MonMaster, utilisé comme identifiant individuel non aisément devinable.

```
chiffrement-pdf.fish ≡ fish
1 cd "$RepertoireTravailSorties"mail_verification_candidats
2 for i in (ls *.tex)
3     lualatex $i
4 end
5 for i in (ls *.pdf)
6     mkdir (basename $i .pdf);
7     qpdf --encrypt (basename $i .pdf) (basename $i .pdf) 256 \
8         -- $i (basename $i .pdf)/dossier_chiffre.pdf
9 end
```

- 36 La fiche est envoyée aux candidat-es avec un courriel d'accompagnement, contenant un lien vers un formulaire permettant aux candidat-es de signaler des incohérences. Ce courriel est enregistré dans un fichier texte mail.txt.

mail.txt ≡

Madame, Monsieur,

Vous avez candidaté au parcours Informatique et droit du Master Droit du numérique de l'Université Paris est Créteil, et nous vous remercions vivement de votre intérêt pour cette formation.

Dans le cadre de la procédure d'examen des candidatures à ce Master, une commission composée de quatre enseignant-es-chercheur-es a été constituée. Cette commission examine, évalue et classe les candidatures en combinant des opérations automatisées, d'une part, et des interventions humaines, d'autre part, selon un processus bien précis décomposé en sept étapes. Un important travail a été mené pour que ce processus soit le plus objectif et le plus transparent possible, ayant bien conscience qu'aujourd'hui, les sélections en Master dans les universités françaises demeurent trop souvent obscures.

Dans ce contexte, nous nous permettons de vous adresser ce message pour aborder deux points.

- 1° *Pour traiter correctement votre candidature, nous aurions besoin de nous assurer de l'exactitude de certaines données, à partir desquelles nous opérons une partie de nos calculs.*

Dans votre cas, les données extraites de votre dossier sur la plateforme nationale MonMaster sont disponibles en pièce jointe, dans un fichier PDF chiffré. Le mot de passe pour y accéder est votre numéro de candidat sur la plateforme MonMaster (ce mot de passe est de la forme CAND suivi de 8 caractères alphanumériques). Si vous venez à constater que ces informations sont incohérentes avec celles que vous avez remplies, vous pouvez signaler cette incohérence sur le formulaire suivant: [URL] (vous avez jusqu'au jeudi 12 avril 15h pour y accéder). Ce formulaire ne doit en aucune façon servir à ajouter de nouvelles informations, mais seulement à vérifier la prise en compte des

informations déjà inscrites. Si les informations sont cohérentes avec celles que vous avez remplies, vous n'avez rien à faire.

- 2° *Par ailleurs, nous souhaitons vous rappeler que, conformément à la législation en vigueur, nous traitons vos données à caractère personnel, sur le fondement de l'exécution d'une mission d'intérêt public au sens du e du 1 de l'article 6 du règlement (UE) 2016/679 du Parlement européen et du Conseil du 27 avril 2016 sur la protection des données (RGPD). Les données traitées lors des opérations automatisées de notre processus d'examen sont les suivantes, telles qu'elles nous sont transmises par la plateforme nationale MonMaster : nom, prénom, numéro de candidat.e ; type et spécialités du baccalauréat ou équivalent ; notes, mention et établissement de préparation des cinq ou six semestres de licence ; éventuelles mentions de master.*

Le responsable de ce traitement est l'université Paris-Est Créteil (UPEC, 61 avenue du général de Gaulle, 94010 Créteil cedex), même si ce traitement est concrètement mis en œuvre par la commission d'examen des candidatures, qui agit dans le cadre de ses attributions propres et dispose d'une autonomie dans l'appréciation des candidatures et l'établissement du classement. Ce traitement a été conçu et développé entièrement en interne à l'UPEC, et plus précisément par les membres de la commission d'examen des candidatures. Il repose sur un programme écrit dans le langage Catala, intégrant des règles explicitement formalisées.

Ont accès aux données à caractère personnel enregistrées dans le traitement, dans la stricte limite de leurs attributions et du besoin d'en connaître, les seuls membres (quatre personnes) de la commission d'examen des candidatures du Master Droit du numérique – parcours Informatique et Droit de l'université Paris XII (université Paris-Est Créteil, UPEC). Ces quatre membres constituent le jury souverain chargé de l'établissement du classement des candidatures. En dehors de ces membres, aucun autre accès aux données à caractère personnel n'est, par principe, autorisé. Peuvent toutefois, à titre exceptionnel et dans la stricte limite de l'exercice de leurs compétences, être amenés à accéder à tout ou partie des données à caractère personnel enregistrées dans le traitement, lorsque cela est nécessaire à l'exercice de leurs missions : 1° les agents habilités du service contentieux de l'université, exclusivement en cas de contentieux relatif au traitement ; 2° l'ingénieur système de la faculté des sciences et technologie de l'université en charge de la gestion des serveurs informatiques internes du laboratoire LACL, dans le cadre de ses seules missions de maintenance et de sécurité des systèmes d'information.

Les données traitées sont conservées pendant une durée de six mois à compter de l'établissement du classement final par la commission d'examen des candidatures, puis supprimées de manière sécurisée (sauf lorsqu'un recours administratif ou contentieux a été engagé, nécessitant leur conservation jusqu'à l'issue de la

procédure).

Vous pouvez accéder aux données vous concernant et exercer vos droits d'accès, de rectification, de limitation que vous tenez des articles 15, 16 et 18 du RGPD à l'adresse suivante : dpo@u-pec.fr. Vous pouvez exercer votre droit d'opposition prévu à l'article 21 du RGPD à la même adresse.

Pour toute question concernant le traitement de vos données à caractère personnel, vous pouvez contacter la déléguée à la protection des données de l'UPEC :

- *via l'adresse électronique : dpo@u-pec.fr ;*
- *ou par courrier postal à l'attention de la déléguée à la protection des données, Université Paris-Est Créteil, 61, avenue du général de Gaulle, 94010 Créteil CEDEX.*

Si vous estimez que, même après avoir introduit une réclamation auprès de l'UPEC, vos droits en matière de protection des données à caractère personnel ne sont pas respectés, vous avez la possibilité d'introduire une réclamation auprès de la CNIL à l'adresse suivante : 3 Place de Fontenoy – TSA 80715 – 75334 Paris CEDEX 07.

Dans le cadre de l'exercice de vos droits, vous devez justifier de votre identité par tout moyen. En cas de doute sur votre identité, les services chargés du droit d'accès et la déléguée à la protection des données se réservent le droit de vous demander les informations supplémentaires qui leur apparaissent nécessaires, y compris la photocopie d'un titre d'identité portant votre signature.

Restant à votre disposition pour toute question, Nous vous prions de croire en nos meilleures salutations,

37 Ce courriel est envoyé par l'opérateur du traitement à chaque candidat·e.

```

courriel-verification.el ≡ emacs-lisp
1  <SIGNATUREMAIL 43>
2  (let (message-mail-user-agent)
3    (dolist
4      <LECTUREFICHIERCANDIDATS 39>
5      (let ((form-letter
6            (lambda ()
7              (save-excursion
8                (goto-char (- (point-max) 45))
9                <PIECEJOINTE 40>
10               <CONTENUMAIL 41>
11               )))
12      (unwind-protect
13        (progn
14          (add-hook 'message-setup-hook form-letter)
15          <TITREMAIL 42>
16          )
17      (remove-hook 'message-setup-hook form-letter))))))

```

- 38 On crée un fichier csv des candidat-es à qui envoyer le courriel, séparé par des points virgules, dont la première colonne est le numéro de candidat-e et la deuxième l'adresse électronique.

```

⟨PRODUCTIONVÉRIFICATIONNOMINAUX 34⟩ +≡ OCaml
1 let () = écriture_csv_travail
2     "mail_verification_candidats/candidats.csv"
3     []
4     (List.map
5         (λc → match c with
6             numero,{adresse_email}
7             → [numero;adresse_email])
8     (List.filter
9         (λx → tri_dossier (snd x) = Nominal)
10        candidats))

```

- 39 Ce fichier csv est ensuite lu pour l'envoi.

```

⟨LECTUREFICHIERCANDIDATS⟩ ≡ emacs-lisp
1 (candidat (cdr (nbutlast
2             (with-temp-buffer
3             (insert-file-contents
4             (concat ⟨REPertoireTRAVAILSORTIES 188⟩
5
6                 ↪ "mail_verification_candidats/candidats.csv"))
7             (split-string (buffer-string) "\n"))))
8 (cl-structuring-bind (numero email) (split-string candidat ";"))

```

```

40 ⟨PIÈCEJOINTE⟩ ≡ emacs-lisp
1 (insert "<#part type=\"application/pdf\" filename=\"\"
2         ⟨REPertoireTRAVAILSORTIES 188⟩ "mail_verification_candidats/"
3         numero "/dossier_chiffre.pdf\"
4         ↪ disposition=attachement><#/part>\n")

```

```

41 ⟨CONTENUMAIL⟩ ≡ emacs-lisp
1 (insert-file-contents (concat ⟨REPertoireENTREES 186⟩
2                             "../mails_1_candidats/mail.txt"))

```

```

42 ⟨TITREMAIL⟩ ≡ emacs-lisp
1 (message-mail email
2     "[Master droit du numérique UPEC] Poursuite de la
3     ↪ procédure de candidature")

```

```

43 ⟨SIGNATUREMAIL⟩ ≡ emacs-lisp
1 (setq message-signature
2     "La commission d'examen des candidatures")

```

3.1.2 Correction

- 44 Lorsqu'un-e candidat-e a formé une contestation et que celle-ci montre bien que le programme s'exécutait sur des données inexactes, ces données sont mises à jour au moyen de corrections explicites: l'opérateur-ice de traitement reprend alors la ligne correspondante du fichier initial et la modifie manuellement dans un fichier `corrections.csv` situé dans le répertoire de travail des entrées.

```

<LECTUREVÉRIFICATIONNOMINAUX> ≡ OCaml
1 let corrections = construction_candidats
2   ( <REPertoireTRAVAILENTRÉES 189> ^
   ↪ "corrections.csv")

```

Étendu en §46.

- 45 <FONCTIONS CALCUL> ≡ OCaml
- ```

1 let rec corrige candidats corrections =
2 match corrections with
3 [] → candidats
4 | (clef,candidat)::q
5 → (clef,candidat)::(corrige (List.remove_assoc clef candidats)
 ↪ q)

```

Étendu en §60, 71, 132, 152, 153, 252, 269.

- 46 <LECTUREVÉRIFICATIONNOMINAUX 44> + ≡ OCaml
- ```

1 let candidats = corrige candidats corrections

```

3.2 Calcul sur les notes de licence

- 47 Les notes de licence sont constituées d'une note pour chaque semestre validé, correspondant à la moyenne générale du semestre en question. Une note est ainsi renseignée pour chacun des cinq premiers semestres, et le cas échéant pour le sixième. Si un même semestre a été passé plusieurs fois, c'est la dernière note qui compte: un redoublement efface l'année redoublée (voir § 250). La note du semestre 6 est présente dans le cas d'un-e candidat-e ayant déjà validé une première L3 ou redoublé.

```

<NOTESLICENCE> ≡ Catala
1 déclaration structure NotesLicence:
2   donnée note_semestre contenu
3     (décimal, décimal, décimal, décimal, décimal,
4     optionnel de décimal)

```

- 48 Le calcul du score de la première phase ne dépend que de ces notes.

```

<SCOREPHASELICENCE> ≡ Catala
1 déclaration champ d'application ScoreSemestres:
2   entrée notes contenu NotesLicence
3   résultat points_semestres contenu décimal

```

- 49 Le calcul du score de licence est fondé sur une moyenne arithmétique des notes de licence — c'est-à-dire des moyennes générales obtenues à chaque semestre validé —, pondérées par des coefficients. Ces coefficients ont pour objet de valoriser les parcours ascendants, en accordant un poids plus important au semestre ou aux semestres de L3.

$\langle \text{SCORELICENCE} \rangle \equiv$ Catala

```

1  déclaration moyenne_pondérée contenu décimal
2  dépend de
3      notes contenu NotesLicence,
4      coefficients contenu
5      (décimal, décimal, décimal, décimal,
6      décimal, décimal)
7  égal à
8      ( coefficients.1 × notes.note_semestre.1
9      + coefficients.2 × notes.note_semestre.2
10     + coefficients.3 × notes.note_semestre.3
11     + coefficients.4 × notes.note_semestre.4
12     + coefficients.5 × notes.note_semestre.5
13     + (selon notes.note_semestre.6 sous forme
14         -- Présent contenu note:
15         coefficients.6 × note
16         -- Absent: 0,0))
17     /
18     ( coefficients.1 + coefficients.2
19     + coefficients.3 + coefficients.4
20     + coefficients.5
21     + (selon notes.note_semestre.6 sous forme
22         -- Présent contenu note:
23         coefficients.6
24         -- Absent: 0,0))

```

Étendu en §50, 52.

- 50 Cette moyenne arithmétique pondérée est ensuite transformée par une fonction d'écartement ayant les propriétés suivantes :

croissance le score calculé sur une moyenne pondérée est supérieur sur une moyenne supérieure ;

continuité il n'y a pas d'effet de seuil dans le calcul ;

borne le score est borné (il y a un score minimal et un score maximal) ;

constance au-dessus d'une certaine note au dessus d'une certaine moyenne pondérée (14), le score est constant ;

linéarité au-dessous d'une certaine note au dessous d'une certaine moyenne pondérée (10,5), le score est linéaire ;

concavité entre ces deux bornes entre 10,5 et 14, la fonction est concave : l'augmentation du score est de plus en plus faible à mesure que la moyenne augmente.

$\langle \text{SCORELICENCE 49} \rangle + \equiv$ Catala

1 $\langle \text{FONCTIONSÉPARATION 56} \rangle$

- 51 Le calcul de cette fonction utilise une puissance, fonction absente de la bibliothèque mathématique standard de CATALA. Elle a donc été programmée dans un module mathématique (voir Annexe 195).

⟨DÉPENDANCES⟩ ≡ CATALA

1 > **Usage de Math**

- 52 Le score est calculé en faisant la moyenne pondérée des notes des semestres effectivement validés, et en appliquant la fonction écartant les notes à cette moyenne.

⟨SCORELICENCE 49⟩ + ≡ CATALA

```
1 champ d'application ScoreSemestres:
2   définition points_semestres égal à
3     écartement de
4     (moyenne_pondérée de notes, coefficient_semestres)
```

- 53 Le score est monotone : un·e candidat·e ayant une moyenne supérieure ou égale à celle d'un·e autre candidat·e aura un score supérieur ou égal.
- 54 Le score obtenu à cette phase est inférieur ou égal à `maximum_score_notes_licence`.

3.2.1 Coefficients et paramètres

- 55 Les coefficients de la moyenne arithmétique pondérée sont tels que les deux années de L1 et L2 comptent autant, et que l'année de L3, si elle est entière, compte deux fois plus.

⟨COEFFICIENTS⟩ ≡ CATALA

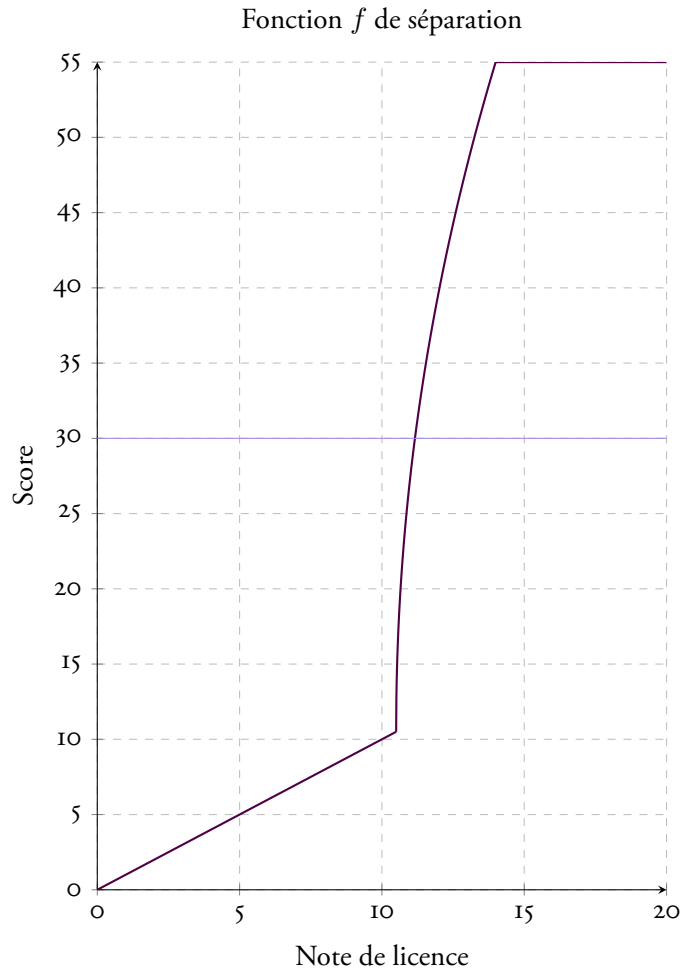
```
1 déclaration coefficient_semestres contenu
2   (décimal, décimal, décimal, décimal, décimal, décimal)
3   égal à ((1,0), (1,0), (1,0), (1,0), (1,5), (1,5))
```

- 56 La fonction de séparation est donnée par :

$$f : [0; 20] \rightarrow [0; 55]$$

$$x \mapsto \begin{cases} x & \text{si } x \leq 10,5; \\ 10,5 + 23,950 \times (x - 10,5)^{0,495} & \text{sinon.} \end{cases}$$

Elle peut être représentée par :



$\langle \text{FONCTIONSÉPARATION} \rangle \equiv$

Catala

```

1  déclaration écartement contenu décimal
2  dépend de note contenu décimal
3  égal à
4  soit n égal à
5     si note < 10,5
6     alors note
7     sinon
8     soit m égal à
9     (note - 10,5)
10    dans 10,5 + 23,950 × Math.pow de m, (0,495)
11    dans Décimal.min de n, (décimal de maximum_score_notes_licence)

```

57 Le score maximal (évoqué en § 54) est fixé à 55.

$\langle \text{COEFFICIENTS 55} \rangle + \equiv$

Catala

```

1  déclaration maximum_score_notes_licence contenu entier égal à 55

```

3.3 Traitement des candidatures dont le score est inférieur à un seuil

58 En dessous d'un seuil sur le score (et donc sur les notes de licence), les candidatures relevant du groupe nominal sont considérées comme présentant un niveau insuffisant pour que leur examen soit poursuivi.

$\langle \text{PRODUCTIONSOUSSEUIL} \rangle \equiv$ OCaml

```

1 let nominaux = List.filter
2     ( $\lambda x \rightarrow \text{tri\_dossier (snd x) = Nominal}$ )
3     candidats

```

Étendu en §61, 62.

- 59 La commission choisit ce seuil sur le score obtenu par les candidat-es à l'issue du calcul du score.

$\langle \text{SEUIL} \rangle \equiv \dots$ OCaml

- 60 La fonction

```

1 val notes_licence_of_candidat :
2     dossier_formulaire  $\rightarrow$  ScoreSemestres_in.t

```

applique la fonction ScoreSemestres ci-dessus.

$\langle \text{FONCTIONS CALCUL 45} \rangle + \equiv$

```

1  $\langle \text{MISEENFORME 260} \rangle$ 

```

- 61 Deux documents de travail sont produits :

- un fichier csv contenant une ligne par candidat-e, et comme colonne les noms, prénoms et numéro de candidat-es, ainsi qu'une colonne « Oui/Non » ;
- un fichier pdf contenant, dans le même ordre, l'intégralité du dossier (formulaire et pièces) fournis par les candidat-es.

$\langle \text{PRODUCTIONSOUSSEUIL 58} \rangle + \equiv$ OCaml

```

1 let candidats_sous_seuil =
2     List.filter
3     ( $\lambda c \rightarrow$ 
4         match (score_semestres (notes_licence_of_candidat (snd c)))
5             with {points_semestres}
6                  $\rightarrow$  Q.to_float points_semestres  $\leq$  seuil)
7     nominaux

```

- 62 $\langle \text{PRODUCTIONSOUSSEUIL 58} \rangle + \equiv$ OCaml

```

1 let () = ecriture_csv_travail
2     "candidats_sous_seuil.csv"
3     ["Nom"; "Prénom"; "Numéro"; "Réintégration"]
4     (List.map
5         ( $\lambda c \rightarrow$  match c with
6             numero, {nom; prenom}
7                  $\rightarrow$  [nom; prenom; numero])
8     candidats_sous_seuil)

```

- 63 La commission désigne en son sein un-e rapporteur-ice chargé-e de lire les dossiers des candidat-es dont le score est inférieur au seuil fixé au § 59. Sur la base de son rapport, et en cas de circonstances exceptionnelles, la commission peut décider de réintégrer certaines candidatures dans le processus d'évaluation. Cette décision donne lieu à la production d'un document de travail indiquant, pour chaque candidature concernée, si elle est réintégrée (« Oui ») ou (« Non »).

64 La fonction `lecture_csv_travail`. $\langle \text{DÉCLARATION FONCTIONS LECTURE} \rangle \equiv \overline{\text{OCaml}}$

```
1 val lecture_csv_travail : string → string list list
```

65 La liste `rattrapes` contient les numéros des candidatures rattrapées.

$\langle \text{LECTURE SOUS SEUIL} \rangle \equiv \overline{\text{OCaml}}$

```
1 let rattrapes =
2   List.map
3     (λ ligne →
4       match ligne with
5         _::_:n::_ → n
6         | _ →
7           ⟨LIGNE INVALIDE 262⟩
8     )
9   (List.filter (λ x →
10     match x with
11       _::_:r::_ → r = "Oui"
12       | _ → false)
13     (lecture_csv_travail "candidats_sous_seuil.csv"))
```

Étendu en §66.

66 La liste d'association `candidats_sous_seuil` associe son dossier à chaque numéro de candidatures sous le seuil et n'ayant pas été rattrapée.

$\langle \text{LECTURE SOUS SEUIL 65} \rangle + \equiv \overline{\text{OCaml}}$

```
1 let candidats_sous_seuil =
2   List.fold_left
3     (λ cc → λ x → List.remove_assoc x cc)
4     candidats_sous_seuil
5     rattrapes
```

3.4 Appréciation des candidatures dont le score est supérieur du seuil ou qui ont été rattrapées

67 Les candidatures dont le score est supérieur au seuil, ainsi que celles qui ont été rattrapées, sont considérées comme présentant un niveau académique suffisant. La commission désigne en son sein un·e rapporteur·ice chargé·e d'examiner leur lettre de motivation et leur parcours. Sur la base de ce rapport, la commission attribue deux notes, chacune comprise entre 0 et 10 :

- une note d'appréciation de la lettre de motivation ;
- une note d'appréciation du dossier.

$\langle \text{APPRÉCIATION} \rangle \equiv \overline{\text{Catala}}$

```
1 donnée appreciation_lettre contenu entier
2 donnée appreciation_dossier contenu entier
```

68 Deux documents de travail sont produits :

- un fichier csv contenant une ligne par candidat·e, et comme colonne les noms, prénoms et numéro de candidat·es, ainsi que deux colonnes « Lettre » et « Dossier » ;

— un fichier PDF contenant, dans le même ordre, l'intégralité du dossier (formulaire et pièces) fournis par les candidat-es.

⟨PRODUCTIONSEUIL⟩ ≡ OCaml

```

1 let candidats_sur_seuil =
2   List.filter
3     (λc →
4       (match (score_semestres (notes_licence_of_candidat (snd c)))
5         with {points_semestres} → Q.to_float points_semestres >
6           ⇨ seuil)
7       || List.mem (fst c) rattrapes
8     )
9   nominaux

```

Étendu en §69.

69 ⟨PRODUCTIONSEUIL 68⟩ +≡ OCaml

```

1 let () =
2   ecriture_csv_travail "candidats_seuil.csv"
3     ["Nom"; "Prénom"; "Numéro";
4     "Appréciation Lettre"; "Appréciation Dossier"]
5   (List.map (λc →
6     match c with
7       numero, {nom; prenom}
8       → [nom; prenom; numero])
9     candidats_sur_seuil)

```

70 Sur la base du rapport mentionné au § 67, la commission attribue les notes de chaque candidature en produisant un document de travail dans lequel elle a rempli les deux colonnes.

⟨LECTURESEUIL⟩ ≡ OCaml

```

1 let notes_appreciations =
2   List.map
3     (λligne →
4       match ligne with
5         _::_:::numero::note1::note2::_ →
6         (numero, int_of_string note1, int_of_string note2)
7       | _ →
8         ⟨LIGNE|INVALIDE 262⟩
9     )
10  (lecture_csv_travail "candidats_seuil.csv")

```

Étendu en §72, 73.

71 ⟨FONCTIONS_CALCUL 45⟩ +≡ OCaml

```

1 let appreciations_candidats candidats notes =
2   let rec appreciations_candidats
3     candidats candidats_apprecies notes =
4     match notes with
5     [] → let () = assert (candidats = []) in
6     candidats_apprecies

```

```

7 | (clef,note_lettre,note_dossier)::q →
8   try (let c = List.assoc clef candidats in
9     appreciations_candidats
10    (List.remove_assoc clef candidats)
11    ((clef,c,
12     construit_candidat_final c note_lettre note_dossier)::
13     candidats_apprecies)
14    q) with
15 | Not_found → failwith clef
16 in appreciations_candidats candidats [] notes

```

72 $\langle \text{LECTURESEUIL } 70 \rangle + \equiv$ OCaml

```

1 let candidats_score_tri =
2   List.sort
3   (λ(_,(_,_,x)) → λ(_,(_,_,y)) → Float.compare y x)
4   (List.map
5     (λ(clef,c,cc)
6       → let ccc = score_candidat {c_in = cc}
7         in (clef,(c,ccc,match ccc with
8             {score} → Q.to_float score)))
9     (appreciations_candidats candidats_sur_seuil
10    ↪ notes_appreciations))

```

73 $\langle \text{LECTURESEUIL } 70 \rangle + \equiv$ OCaml

```

1 let candidats_tri =
2   List.map (λ(clef,(c,_,_)) → (clef,c)) candidats_score_tri

```

74 Ces notes servent à calculer un score de la candidature.

$\langle \text{SCOREDOSSIER} \rangle \equiv$ Catala

```

1 déclaration champ d'application ScoreDossier:
2   entrée appreciation_lettre contenu entier
3   entrée appreciation_dossier contenu entier
4   résultat points_dossier contenu entier

```

75 Ce score est calculé en faisant la somme de chaque note.

$\langle \text{SCOREDOSSIER } 74 \rangle + \equiv$ Catala

```

1 champ d'application ScoreDossier:
2   définition points_dossier égal à
3     appreciation_lettre + appreciation_dossier

```

76 Le score maximal pouvant être atteint par une candidature est donc égal à 20.

$\langle \text{COEFFICIENTS } 55 \rangle + \equiv$ Catala

```

1 déclaration score_dossier_max contenu entier égal à 20

```

3.5 Calcul sur l'intégralité du dossier

- 77 La phase de calcul du score final des candidatures relevant du groupe nominal produit non seulement un score global, mais aussi les scores intermédiaires obtenus dans chaque catégorie, afin de permettre leur affichage dans la notice individuelle d'information transmise à chaque candidat·e.

⟨SCORECOMPLETNOMINAL⟩ ≡ Catala

```

1  déclaration champ d'application ScoreCandidat:
2  entrée c contenu CandidatCompletNominal
3  ⟨SCORESINTERMÉDIAIRES 81⟩
4  résultat score contenu décimal
```

- 78 Le processus de sélection est monotone: pour chaque élément entrant en compte dans le calcul du score, l'améliorer améliore le score.

- 79 Le programme ne dépend que des données suivantes:
- les caractéristiques (type, série, spécialité) du baccalauréat des candidat·es. En particulier, les notes du baccalauréat ne comptent pas;
 - la note de chaque semestre de licence. En particulier, on ne recalcule pas de note augmentant le poids de certaines matières, et on ne regarde pas les majeures ou mineurs choisies;
 - le nombre d'années de master déjà validées et le fait d'avoir obtenu ou non une licence distincte de celle au titre de laquelle la candidature est déposée;
 - le fait que la licence a été préparée à l'UPEC, ce qui s'apprécie selon l'établissement d'inscription de la troisième année de licence. En particulier, toutes les autres licences de droit d'universités publiques françaises sont considérées équivalentes: on ne cherche pas à en redresser les notes;
 - l'appréciation donnée par la commission en fonction du dossier.

⟨CANDIDATE⟩ ≡ Catala

```

1  déclaration structure CandidatCompletNominal:
2  ⟨CARACTÉRISTIQUEBAC 83⟩
3  donnée notes contenu NotesLicence
4  ⟨GRADEUNIVERSITAIRE 107⟩
5  ⟨UPEC 116⟩
6  ⟨APPRÉCIATION 67⟩
```

- 80 Le calcul est divisé en parties calculant respectivement le score lié au baccalauréat, aux notes de licence (déjà présenté en Section 3.2), aux autres diplômes nationaux, au dossier (Section 3.4), et à l'établissement, puis en les réunissant en un score total.

⟨CALCULS⟩ ≡ Catala

```

1  ⟨SCOREBAC 84⟩
2  ⟨SCORELICENCE 49⟩
3  ⟨SCOREAUTREDIPLÔMENATIONAL 108⟩
4  ⟨SCOREDOSSIER 74⟩
5  ⟨SCOREÉTABLISSEMENT 117⟩
6  ⟨SCORECANDIDAT 121⟩
```

81 `<SCORESINTERMÉDIAIRES> ≡` Catala

1 résultat points_bac contenu décimal

2 résultat points_licence contenu décimal

3 résultat points_diplômes contenu décimal

4 résultat points_dossier contenu décimal

5 résultat points_établissement contenu décimal

3.6 Baccalauréat

82 Il y a trois types de baccalauréats, définis respectivement aux articles D. 334-I, D. 336-I et D. 337-5I du Code de l'éducation, et ayant chacun des caractéristiques. Le présent programme est applicable aux candidat·es ayant obtenu un baccalauréat français, quelque soit le type du diplôme de baccalauréat. On distingue ceux ayant obtenu leur baccalauréat général avant l'année scolaire 2020 → 2021 (et donc des séries d'orientation) de ceux d'après (et donc des spécialités).

`<TYPESBACCALAURÉATS> ≡` Catala

1 `<TYPESÉRIESSPÉCIALITÉS 86>`

2 déclaration énumération **CaractéristiquesBac:**

3 -- **GénéralSpécialités** contenu

 ↔ `<CONTENUBACCALAURÉATGÉNÉRALSPÉCIALITÉS 87>`

4 -- **GénéralSéries** contenu `<CONTENUBACCALAURÉATGÉNÉRALSÉRIES 91>`

5 -- **Technologique** contenu `<CONTENUBACCALAURÉATTECHNOLOGIQUE 95>`

6 -- **Professionnel**

Étendu en §105, 131.

83 La seule information utilisée par le programme est le type de baccalauréat (spécialité, série), pour valoriser le fait d'avoir étudié des disciplines formelles comme les mathématiques ou l'informatique. On ne regarde en particulier ni les notes, ni la mention.

`<CARACTÉRISTIQUEBAC> ≡` Catala

1 donnée baccalauréat contenu **CaractéristiquesBac**

84 Un score lié au baccalauréat du ou de la candidat·e est calculé.

`<SCOREBAC> ≡` Catala

1 déclaration champ d'application **ScoreBac:**

2 entrée baccalauréat contenu **CaractéristiquesBac**

3 résultat points_bac contenu entier

Étendu en §88, 89, 92, 93, 96, 97, 98, 106, 135, 136.

85 Le nombre de points obtenus est inférieur ou égal à `maximum_score_baccalauréat`.

3.6.1 Baccalauréat général

86 Les spécialités du baccalauréat général sont données dans l'annexe de l'arrêté du 16 juillet 2018 susvisé. Il s'agit de:

- Arts
- Biologie-écologie

- Education physique, pratiques et culture sportives
- Histoire-géographie, géopolitique et sciences politiques
- Humanités, littérature et philosophie
- Langues, littératures et cultures étrangères et régionales
- Littérature et langues et cultures de l'antiquité
- Mathématiques
- Numérique et sciences informatiques
- Physique-chimie
- Sciences de la vie et de la Terre
- Sciences de l'ingénieur
- Sciences économiques et sociales

⟨TYPESÉRIESSPÉCIALITÉS⟩ ≡

Catala

```

1  déclaration énumération SpécialitéGénéral:
2    -- Arts
3    -- BiologieÉcologie
4    -- EducationPhysiquePratiquesCultureSportives
5    -- HistoireGéographieGéopolitiqueEtSciencesPolitiques
6    -- HumanitésLittératurePhilosophie
7    -- LanguesLittératuresEtCulturesÉtrangères
8    -- LittératuresLanguesEtCulturesAntiquité
9    -- Mathématiques
10   -- NumériqueEtSciencesInformatiques
11   -- PhysiqueChimie
12   -- SciencesDeLaVieEtDeLaTerre
13   -- SciencesDeL'ingénieur
14   -- SciencesÉconomiquesEtSociales

```

87 On ne considère que les spécialités de Terminale, donc deux spécialités pour un baccalauréat général. La spécialité abandonnée en Première est ignorée.

⟨CONTENUBACCALAURÉATGÉNÉRALSPÉCIALITÉS⟩ ≡

Catala

```

1  (SpécialitéGénéral, SpécialitéGénéral)

```

88 Pour chaque spécialité, un certain nombre de points est donné.

⟨SCOREBAC 84⟩ + ≡

Catala

```

1  déclaration champ d'application PointsBacGénéral:
2    entrée spécialité contenu SpécialitéGénéral
3    résultat points_specialite contenu entier

```

89 Le score est calculé en additionnant le nombre de points de chacune des deux spécialités, dans la limite de maximum_score_baccalauréat.

⟨SCOREBAC 84⟩ + ≡

Catala

```

1  champ d'application ScoreBac:
2    définition points_bac
3    sous condition baccalauréat sous forme GénéralSpécialités
4    conséquence égal à
5    minimum de

```

```

6      [
7      maximum_score_baccalauréat;
8      selon baccalauréat sous forme
9      -- GénéralSpécialités contenu spécialités:
10     (résultat de PointsBacGénéral avec {
11     -- spécialité: spécialités.1
12     }.points_specialite) +
13     (résultat de PointsBacGénéral avec {
14     -- spécialité: spécialités.2
15     }.points_specialite)
16     -- n'importe quel: 0
17     ]

```

90 Jusqu'en 2020, le baccalauréat général avait trois séries.

⟨TYPESÉRIESSPÉCIALITÉS 86⟩ + ≡

Catala

```

1  déclaration énumération SérieGénéral:
2  -- Scientifique
3  -- ÉconomiqueEtSocial
4  -- Littéraire

```

91 On ne considère que la série, et pas les éventuelles options.

⟨CONTENUBACCALAURÉATGÉNÉRALSÉRIES⟩ ≡

Catala

```

1  SérieGénéral

```

92 Pour chaque série, un certain nombre de points est donné.

⟨SCOREBAC 84⟩ + ≡

Catala

```

1  déclaration champ d'application PointsBacGénéralSérie:
2  entrée série contenu SérieGénéral
3  résultat points_série_général contenu entier

```

93 Le score est calculé en considérant seulement le nombre de points de la série.

⟨SCOREBAC 84⟩ + ≡

Catala

```

1  champ d'application ScoreBac:
2  définition points_bac
3  sous condition baccalauréat sous forme GénéralSéries
4  conséquence égal à
5  selon baccalauréat sous forme
6  -- GénéralSéries contenu série:
7  (résultat de PointsBacGénéralSérie avec {
8  -- série: série
9  }.points_série_général)
10 -- n'importe quel: 0

```

3.6.2 Baccalauréat technologique

- 94 Les séries du baccalauréat technologique sont définies à l'article D. 336-3 du code de l'éducation :

« Le baccalauréat technologique comprend les séries suivantes :

- 1° Série ST2S : sciences et technologies de la santé et du social ;
- 2° Série STI2D : sciences et technologies de l'industrie et du développement durable ;
- 3° Série STL : sciences et technologies de laboratoire ;
- 4° Série STMG : sciences et technologies du management et de la gestion ;
- 5° Série STAV : sciences et technologies de l'agronomie et du vivant ;
- 6° Série STHR : sciences et technologies de l'hôtellerie et de la restauration ;
- 7° Série S2TMD : sciences et techniques du théâtre, de la musique et de la danse ;
- 8° Série STD2A : sciences et technologies du design et des arts appliqués.

Le baccalauréat technologique comprend des épreuves ou des évaluations de contrôle continu portant sur les enseignements communs dispensés à tous les élèves et les enseignements de spécialité suivis par l'élève ainsi que, le cas échéant, sur des enseignements optionnels. »

(TYPESÉRIESSPÉCIALITÉS 86) + ≡

Catala

- 1 déclaration énumération **SérieTechnologique** :
- 2 -- **ST2S**
- 3 -- **STI2D**
- 4 -- **STL**
- 5 -- **STMG**
- 6 -- **STAV**
- 7 -- **STHR**
- 8 -- **S2TMD**
- 9 -- **STD2A**

- 95 (CONTENUBACCALAURÉATTECHNOLOGIQUE) ≡

Catala

- 1 **SérieTechnologique**

- 96 Pour chaque série, un certain nombre de points est donné.

(SCOREBAC 84) + ≡

Catala

- 1 déclaration champ d'application **PointsBacTechnologique** :
- 2 entrée série contenu **SérieTechnologique**
- 3 résultat points_série_technologique contenu entier

- 97 Le score est calculé en prenant simplement le nombre de points de la série.

(SCOREBAC 84) + ≡

Catala

- 1 champ d'application **ScoreBac** :
- 2 définition points_bac
- 3 sous condition baccalauréat sous forme **Technologique**

Annexe du décret n° 2006-583 du 23 mai 2006 relatif aux dispositions réglementaires du livre III du code de l'éducation (décrets en Conseil d'État et décrets) ; article 4 du décret n° 2008-263 du 14 mars 2008 relatif aux dispositions réglementaires du livre IV du code de l'éducation (Décrets en Conseil d'État et décrets) ; article 5 du décret n° 2009-553 du 15 mai 2009 relatif aux dispositions réglementaires du livre V du code de l'éducation (Décrets en Conseil d'État et décrets) ; article 1 du décret n° 2010-565 du 27 mai 2010 relatif aux enseignements du second degré des voies générale et technologique ; article 1 du décret n° 2011-1218 du 29 septembre 2011 relatif aux enseignements du second degré des voies générale et technologique et modifiant le code de l'éducation (partie réglementaire, livre III) ; article 2 du décret n° 2015-270 du 11 mars 2015 relatif au baccalauréat technologique série sciences et technologies de l'hôtellerie et de la restauration ; article 19 du décret n° 2018-614 du 16 juillet 2018 modifiant les dispositions du code de l'éducation relatives aux enseignements conduisant au baccalauréat général et aux formations technologiques conduisant au baccalauréat technologique ; article 2 du décret n° 2019-750 du 19 juillet 2019 rénovant le baccalauréat technologique série « techniques de la musique et de la danse » ; article 12 du décret n° 2021-983 du 27 juillet 2021 modifiant les dispositions du code de l'éducation relatives au baccalauréat général et au baccalauréat technologique.

```

4     conséquence égal à
5     selon baccalauréat sous forme
6     -- Technologique contenu série:
7     (résultat de PointsBacTechnologique avec {
8         -- série: série
9     }.points_série_technologique)
10    -- n'importe quel: 0

```

3.6.3 Baccalauréat professionnel

- 98 La plateforme MonMaster ne fournit pas la spécialité des baccalauréats professionnels. De ce fait, les candidatures déposées par des titulaires de tels baccalauréats ne peuvent être traitées intégralement automatiquement (voir § 128).

⟨SCOREBAC 84⟩ + ≡

Catala

```

1  champ d'application ScoreBac:
2  définition points_bac
3  sous condition baccalauréat sous forme Professionnel
4  conséquence égal à
5  selon baccalauréat sous forme
6  -- Professionnel: 0
7  -- n'importe quel: 0

```

- 99 Le diplôme du baccalauréat professionnel est « délivré au titre d'une spécialité professionnelle » (article D. 337-51 du code de l'éducation). Ces spécialités sont créées par arrêté du ou de la ministre chargé-e de l'éducation (éventuellement conjoint avec un-e autre ministre) après avis des commissions professionnelles consultatives compétentes. L'arrêté portant création de la spécialité établit le référentiel des activités professionnelles, le référentiel de certification ainsi que le règlement particulier qui fixe les conditions de délivrance de ce diplôme (article D. 337-53 du même code).

Un score non-nul est attribué aux spécialités suivantes :

- cybersécurité, informatique et réseaux, électronique (anciennement « Systèmes numériques ») ;
- métiers de l'électricité et de ses environnements connectés ;
- maintenance des systèmes de production connectés (anciennement « Maintenance des équipements industriels »).

Arrêté du 13 février 2023 portant création de la spécialité « cybersécurité, informatique et réseaux, électronique » de baccalauréat professionnel et fixant ses modalités de délivrance.

Arrêté du 8 janvier 2024 portant création de la spécialité « Métiers de l'électricité et de ses environnements connectés » de baccalauréat professionnel et fixant ses modalités de délivrance.

Arrêté du 9 décembre 2020 portant création de la spécialité « Maintenance des systèmes de production connectés » de baccalauréat professionnel et fixant ses modalités de délivrance.

3.6.4 Coefficients des baccalauréats

100 Les coefficients attribués aux spécialités et séries des baccalauréats sont donnés par la table de la Figure 1.

101 Pour les baccalauréats généraux, les spécialités considérées comme formant de manière conséquente aux disciplines formelles ont 10 points. Il s'agit de Numérique et sciences informatiques; Mathématiques; Sciences de l'ingénieur.

⟨COEFFICIENTS 55⟩ + ≡ Catala

```

1 champ d'application PointsBacGénéral :
2   définition points_specialite égal à
3     selon spécialité sous forme
4       -- BiologieÉcologie: 6
5       -- Mathématiques: 10
6       -- NumériqueEtSciencesInformatiques: 10
7       -- PhysiqueChimie: 6
8       -- SciencesDeLaVieEtDeLaTerre: 6
9       -- SciencesDeLingénieur: 10
10      -- SciencesÉconomiquesEtSociales: 6
11      -- n'importe quel: 0

```

Étendu en §57, 76, 101, 102, 103, 104, 113, 114, 115, 120, 127.

102 Pour les baccalauréats généraux d'avant 2020, les séries formant de manière conséquente aux disciplines formelles sont valorisées.

⟨COEFFICIENTS 55⟩ + ≡ Catala

```

1 champ d'application PointsBacGénéralSérie :
2   définition points_série_général égal à
3     selon série sous forme
4       -- Scientifique: 15
5       -- ÉconomiqueEtSocial: 10
6       -- Littéraire: 0

```

103 Au vu des volumes horaires, définis respectivement dans l'arrêté du 16 juillet 2018 susvisé, seules trois séries donnent lieu à l'attribution de points. La série « sciences et technologies de l'industrie et du développement durable (STI2D) » donne lieu à l'attribution de 6 points. Les séries « sciences et technologies de laboratoire (STL) » et « sciences et technologies du management et de la gestion (STMG) » donnent lieu à l'attribution de 4 points. Les autres séries ne donnent lieu à aucun point.

⟨COEFFICIENTS 55⟩ + ≡ Catala

```

1 champ d'application PointsBacTechnologique:
2   définition points_série_technologique égal à
3     selon série sous forme
4       -- STI2D: 6
5       -- STL: 4
6       -- STMG: 4
7       -- n'importe quel: 0

```

- 104 Le score maximal obtenu sur le baccalauréat est fixé à 15.

$\langle \text{COEFFICIENTS } 55 \rangle + \equiv$ Catala

1 déclaration maximum_score_baccalauréat contenu entier égal à 15

3.6.5 Équivalences et baccalauréats étrangers

- 105 Dès lors que la capacité en droit ouvre la possibilité de poursuivre des études en licence de droit, conformément à l'article 1^{er} de l'arrêté du 25 septembre 2021 susvisé, les candidatures déposées par des personnes titulaires d'une capacité de droit sont assimilées, pour l'application du présent programme, à des candidatures de titulaires d'un diplôme permettant l'accès à l'université.

$\langle \text{TYPESBACCALAURÉATS } 82 \rangle + \equiv$ Catala

1 -- **Capacité**

- 106 Ce diplôme ne contient pas de sciences formelles.

$\langle \text{SCOREBAC } 84 \rangle + \equiv$ Catala

1 champ d'application **ScoreBac**:
 2 définition points_bac
 3 sous condition baccalauréat sous forme **Capacité**
 4 conséquence égal à 0

3.7 Master et licence

- 107 Outre le baccalauréat, le programme prend en compte d'autres diplômes nationaux de l'enseignement supérieur français détenus ou préparés par les candidat·es. Deux informations sont utilisées: le nombre d'années de master déjà validées et le fait d'avoir obtenu ou non une licence distincte de celle au titre de laquelle la candidature est déposée. L'intitulé du master et celui de la licence ne sont pas pris en compte.

$\langle \text{GRADEUNIVERSITAIRE} \rangle \equiv$ Catala

1 donnée master_années_validées contenu entier
 2 donnée licence_obtenue contenu booléen

Étendu en § .

- 108 Le programme calcule, à partir de ces informations, un score relatif aux autres diplômes nationaux détenus par les candidat·es. Ce score dépend du nombre d'années de master validées et de l'obtention éventuelle d'une autre licence.

$\langle \text{SCOREAUTREDIPLÔMENATIONAL} \rangle \equiv$ Catala

1 déclaration champ d'application **ScoreAutreDiplômeNational**:
 2 entrée années_validées_master contenu entier
 3 entrée licence_obtenue contenu booléen
 4 résultat points_diplômes contenu entier

Étendu en §109, 110, 111.

- 109 Le nombre de points lié aux autres diplômes est nul en général.

$\langle \text{SCOREAUTREDIPLÔMENATIONAL } 108 \rangle + \equiv$ Catala

1 champ d'application **ScoreAutreDiplômeNational:**
 2 définition points_diplômes égal à 0 +
 3 $\langle \text{AUTRELICENCE } 112 \rangle$

110 Si le ou la candidat·e a validé une année de master, le score est égal à score_master1_max.

$\langle \text{SCOREAUTREDIPLÔME NATIONAL } 108 \rangle + \equiv$ Catala

1 champ d'application **ScoreAutreDiplômeNational:**
 2 **exception** définition points_diplômes
 3 sous condition années_validées_master = 1
 4 conséquence égal à score_master1_max +
 5 $\langle \text{AUTRELICENCE } 112 \rangle$

111 S'il a validé une deuxième année de master, le score est égal à score_master2_max.

$\langle \text{SCOREAUTREDIPLÔME NATIONAL } 108 \rangle + \equiv$ Catala

1 champ d'application **ScoreAutreDiplômeNational:**
 2 **exception** définition points_diplômes
 3 sous condition années_validées_master = 2
 4 conséquence égal à score_master2_max +
 5 $\langle \text{AUTRELICENCE } 112 \rangle$

112 Dans les deux cas, le fait d'avoir obtenu une autre diplôme universitaire de niveau licence — distincte de la licence de droit validée ou en cours (s5 ou s6) au titre de laquelle la candidature est déposée, ce qui inclut notamment les doubles licences — ajoute score_autre_licence points supplémentaires.

$\langle \text{AUTRELICENCE} \rangle \equiv$ Catala

1 (si licence_obtenue alors score_autre_licence sinon 0)

113 Le score obtenu en ayant validé une année de master score_master1_max est fixé à 2 points.

$\langle \text{COEFFICIENTS } 55 \rangle + \equiv$ Catala

1 déclaration score_master1_max contenu entier égal à 2

114 Le score obtenu en ayant validé deux années de master score_master2_max est fixé à 5 points.

$\langle \text{COEFFICIENTS } 55 \rangle + \equiv$ Catala

1 déclaration score_master2_max contenu entier égal à 5

115 Le score obtenu en ayant validé une autre licence score_autre_licence est fixé à 5 points.

$\langle \text{COEFFICIENTS } 55 \rangle + \equiv$ Catala

1 déclaration score_autre_licence contenu entier égal à 5

3.8 Établissement d'origine

- 116 Le programme prend en compte l'établissement dans lequel les candidat·es ont préparé leur licence. La seule information utilisée est le fait que l'année de L3 a été validée (ou est en cours de validation) à l'UPEC. En particulier, aucune distinction n'est opérée entre les autres universités françaises.

$\langle \text{UPEC} \rangle \equiv$ Catala

1 donnée candidat_e_UPEC contenu booléen

- 117 Un score lié à l'établissement d'origine du ou de la candidat·e est calculé à partir de cette information.

$\langle \text{SCOREÉTABLISSEMENT} \rangle \equiv$ Catala

1 déclaration champ d'application **ScoreÉtablissement:**

2 entrée licence_UPEC contenu booléen

3 résultat points_établissement contenu entier

- 118 Le score d'établissement d'origine est fixé à zéro en général.

$\langle \text{SCOREÉTABLISSEMENT } 117 \rangle + \equiv$ Catala

1 champ d'application **ScoreÉtablissement:**

2 définition points_établissement égal à 0

- 119 Dans le cas d'un·e candidat·e provenant de l'UPEC, il est fixé à score_établissement.

$\langle \text{SCOREÉTABLISSEMENT } 117 \rangle + \equiv$ Catala

1 champ d'application **ScoreÉtablissement:**

2 **exception** définition points_établissement

3 sous condition licence_UPEC = vrai

4 conséquence égal à score_établissement

- 120 Le score d'établissement est fixé à 5.

$\langle \text{COEFFICIENTS } 55 \rangle + \equiv$ Catala

1 déclaration score_établissement contenu entier égal à 5

3.9 Somme des scores

121 $\langle \text{SCORECANDIDAT} \rangle \equiv$ Catala

1 champ d'application **ScoreCandidat:**

2 définition points_bac égal à

3 décimal de (résultat de **ScoreBac** avec {

4 -- baccalauréat:c.baccalauréat}.points_bac)

122 $\langle \text{SCORECANDIDAT } 121 \rangle + \equiv$ Catala

1 champ d'application **ScoreCandidat:**

2 définition points_licence égal à

3 résultat de **ScoreSemestres** avec {

4 -- notes:c.notes}.points_semestres

123 $\langle \text{SCORECANDIDAT 121} \rangle + \equiv$ Catala

```

1  champ d'application ScoreCandidat:
2  définition points_diplômes égal à
3  décimal de (résultat de ScoreAutreDiplômeNational avec {
4  -- années_validées_master:c.master_années_validées
5  -- licence_obtenue:c.licence_obtenue
6  }.points_diplômes)

```

124 $\langle \text{SCORECANDIDAT 121} \rangle + \equiv$ Catala

```

1  champ d'application ScoreCandidat:
2  définition points_dossier égal à
3  décimal de (résultat de ScoreDossier avec {
4  -- appreciation_lettre:c.appreciation_lettre
5  -- appreciation_dossier:c.appreciation_dossier
6  }.points_dossier)

```

125 $\langle \text{SCORECANDIDAT 121} \rangle + \equiv$ Catala

```

1  champ d'application ScoreCandidat:
2  définition points_établissement égal à
3  décimal de (résultat de ScoreÉtablissement avec {
4  -- licence_UPEC:c.candidat_e_UPEC}.points_établissement)

```

126 Le score total est calculé en additionnant les scores des différents aspects.

$\langle \text{SCORECANDIDAT 121} \rangle + \equiv$ Catala

```

1  champ d'application ScoreCandidat:
2  définition score égal à
3  points_bac
4  + points_licence
5  + points_diplômes
6  + points_dossier
7  + points_établissement

```

127 La somme des coefficients est donc égale à 105.

$\langle \text{COEFFICIENTS 55} \rangle + \equiv$ Catala

```

1  déclaration score_max contenu entier égal à
  ⇨ (maximum_score_notes_licence + score_dossier_max +
  ⇨ maximum_score_baccalauréat + score_établissement +
  ⇨ score_master2_max)

```

points	spécialité générale	séries générales ou technologique	série professionnelle
15		s	
10	Numérique et sciences informatiques Mathématiques Sciences de l'ingénieur	STI2D ES	
6	Physique chimie Sciences de la vie et de la Terre Biologie écologie Sciences économiques et sociales		Cybersécurité, informatique et réseaux, électronique
4		STMG STL	
2			Maintenance des systèmes de production connectés Métiers de l'électricité et de ses environnements connectés

FIGURE 1 : Les coefficients des séries et spécialités des baccalauréats

4 Cas quasi-nominal

- 128 Les candidatures déposées par des personnes titulaires d'un baccalauréat étranger ou d'un baccalauréat français professionnel ne peuvent être traitées uniquement à partir des données du formulaire. Elles relèvent du groupe *quasi-nominal* lorsqu'elles concernent des personnes ayant validé les cinq ou six premiers semestres d'une licence de droit préparée dans une université publique française, comme dans le cas nominal (§ 28).

⟨GROUPESFORMULAIRE 28⟩ + ≡ Catala

1 | **QuasiNominal**

- 129 La commission désigne en son sein un·e rapporteur·ice chargé·e d'évaluer l'équivalence du baccalauréat étranger aux baccalauréats français et de donner une note aux baccalauréats professionnels conformément au tableau de la Figure 1. Cette évaluation vise à permettre le traitement de la candidature dans des conditions comparables à celles du groupe normal.

- 130 Deux documents de travail sont produits :

- un fichier csv contenant une ligne par candidat·e, et comme colonne les noms, prénoms et numéro de candidat·es, ainsi qu'une colonne « Baccalauréat » ;
- un fichier pdf contenant, dans le même ordre, l'intégralité du dossier (formulaire et pièces) fournis par les candidat·es.

⟨PRODUCTIONQUASINOMINAUX⟩ ≡ OCaml

```

1 let quasinominaux =
2   List.filter (λx → tri_dossier (snd x) = QuasiNominal)
3     candidats
4 let () = ecriture_csv_travail "quasinominaux.csv"
5         ["Nom"; "Prénom"; "Numéro"; "baccalaureat"]
6         (List.map
7           (λc → match c with
8             numero, {nom; prenom}
9             → [nom; prenom; numero])
10          quasinominaux)

```

- 131 Sur la base du rapport mentionné au § 129, la commission d'examen des candidatures affecte un certain nombre de points, inspirés des coefficients, aux autres équivalences, y compris les baccalauréats étrangers.

⟨TYPESBACCALAURÉATS 82⟩ + ≡ Catala

```

1 -- Équivalence
2 -- ÉquivalenceÉvaluée contenu entier

```

- 132 ⟨FONCTIONS CALCUL 45⟩ + ≡ OCaml

```

1 let rec equivalences_baccalaureats notes candidats =
2   match notes with
3   [] → candidats
4   | (numero, note)::q →

```

```

5      (numero,
6      {(List.assoc numero candidats) with
7      baccalaureat = CaracteristiquesBac.EquivalenceEvaluee
8      (Z.of_int note)})::
9      (equivalences_baccalaureats q
10     (List.remove_assoc numero candidats))

```

- 133 La commission attribue les notes en produisant un document de travail dans lequel elle a rempli cette colonne.

⟨LECTUREQUASINOMINAUX⟩ ≡ OCaml

```

1  let candidats =
2  equivalences_baccalaureats
3  (List.map
4  (λligne →
5  match ligne with
6  _:::_::numero::note::_ → (numero,int_of_string note)
7  | _ →
8  ⟨LIGNEINVALIDE 262⟩
9  )
10 (lecture_csv_travail "quasinominaux.csv")) candidats

```

Étendu en §134.

- 134 Une fois cette attribution faite, les candidatures relevant du groupe quasi-nominal peuvent être traitées comme celles relevant du groupe nominal, selon la procédure décrite en Section 3.

⟨LECTUREQUASINOMINAUX 133⟩ + ≡ OCaml

```

1  let () = assert
2  (List.filter
3  (λx → tri_dossier (snd x) = QuasiNominal)
4  candidats = [])

```

- 135 ⟨SCOREBAC 84⟩ + ≡ Catala

```

1  champ d'application ScoreBac:
2  définition points_bac
3  sous condition baccalauréat sous forme Équivalence
4  conséquence égal à 0

```

- 136 ⟨SCOREBAC 84⟩ + ≡ Catala

```

1  champ d'application ScoreBac:
2  définition points_bac
3  sous condition baccalauréat sous forme ÉquivalenceÉvaluée
4  conséquence égal à
5  selon baccalauréat sous forme
6  -- ÉquivalenceÉvaluée contenu i: i
7  -- n'importe quel: 0

```

5 Cas nominal avec saisie défaillante

- 137 Certain·es candidat·es, dont les candidatures devraient relever des deux groupes précédents, ont mal rempli le formulaire. Ces candidatures font l'objet d'une instruction spécifique. La commission désigne en son sein un·e rapporteur·ice chargé·e de cette instruction.

$\langle \text{GROUPESFORMULAIRE } 28 \rangle + \equiv$ OCaml

1 | **NominalSaisieDefaillante**

- 138 Deux documents de travail sont produits :

- un fichier csv contenant une ligne par candidat·e, et comme colonne les noms, prénoms et numéro de candidat·es, ainsi qu'une colonne « Rang » ;
- un fichier pdf contenant, dans le même ordre, l'intégralité du dossier (formulaire et pièces) fournis par les candidat·es.

$\langle \text{PRODUCTIONNOMINAUXSAISIEDÉFAILLANTE} \rangle \equiv$ OCaml

```

1 let nominaux_saisie_defaillante =
2   List.filter
3     ( $\lambda x \rightarrow$  tri_dossier (snd x) = NominalSaisieDefaillante)
4     candidats
5 let () = ecriture_csv_travail "nominaux_saisie_defaillante.csv"
6     ["Nom"; "Prénom"; "Numéro"; "Rang"]
7     (List.map
8       ( $\lambda c \rightarrow$  match c with
9         numero, {nom; prenom}  $\rightarrow$ 
10         $\hookrightarrow$  [nom; prenom; numero])
      nominaux_saisie_defaillante)

```

- 139 Sur la base du rapport mentionné § 137, la commission classe les candidatures relevant de ce groupe en attribuant un rang de classement à certaines d'entre elles. Plus précisément, pour chaque candidature, elle remplit soit « Non » (si elle ne souhaite pas la classer), soit le numéro de candidat·e de la personne immédiatement au-dessous (si elle souhaite l'interclasser dans la liste produite), soit un rang (si elle souhaite la classer à la fin de la liste). Elle le fait en produisant un document de travail dans lequel elle a rempli la colonne « Rang ».

$\langle \text{LECTURENOMINAUXSAISIEDÉFAILLANTE} \rangle \equiv$ OCaml

```

1 let resultat_nominaux_saisie_defaillante_inter,
2   resultat_nominaux_saisie_defaillante_apres,
3   resultat_nominaux_saisie_defaillante_non =
4   lecture_liste_complementaire
5   "nominaux_saisie_defaillante.csv" nominaux_saisie_defaillante

```

- 140 La lecture d'un tel fichier est fait par la fonction `lecture_liste_complementaire` qui, à partir d'un nom de fichier csv au format ci-dessus, et de la liste principale des candidatures, produit trois listes: celle des candidatures destinées à être interclassées avec la liste principale, celle des candidatures destinées à être classées après, et celle des candidatures dont l'admission est refusée.

$\langle \text{DÉCLARATIONFONCTIONSLECTURE } 64 \rangle + \equiv$ OCaml

```

1  val lecture_liste_complementaire :
2    string
3    → (string × dossier_formulaire) list
4    → (string × string) list
5      × (string × dossier_formulaire) list
6      × (string × dossier_formulaire) list

```

```

141 <LECTURELISTE_COMPLEMENTAIRE> ≡ OCaml
1  let lecture_liste_complementaire nom liste =
2    let (inter, apres, non) =
3      List.fold_left
4        (λ(inter, apres, non) →
5          λligne →
6            match ligne with
7              _::_:numero::rang::_ →
8                (match rang with
9                  "Non" → (inter, apres, numero::non)
10                 | _ → try (let r = int_of_string rang in
11                           ↪ inter, (numero, r)::apres, non)
12                           with int_of_string →
13                             ↪ (numero, rang)::inter, apres, non)
14                 | _ →
15                   <LIGNE_INVALIDE 262>
16                 )
17                 ([], [], []))
18      (lecture_csv_travail nom)
19      in (inter, List.map (λx → (fst x), List.assoc (fst x) liste)
20          (List.sort (λa → λb → Int.compare (snd a) (snd b))
21            ↪ apres),
22          List.map (λx → x, List.assoc x liste) non)

```

6 Cas des autres licences de droit

142 Les candidatures déposées par des personnes ayant préparé leur licence de droit ailleurs que dans une université publique française — y compris pour ce qui concerne les établissements d'enseignement supérieur privés d'intérêt général — ne sont pas directement comparables à celles relevant des autres groupes, au regard des données utilisées par le programme. Afin de tenir compte des spécificités de ces situations et d'assurer une évaluation adaptée, la commission désigne en son sein un·e rapporteur·ice chargé·e d'instruire les dossiers en question.

```

<GROUPES_FORMULAIRE 28> + ≡ OCaml
1 | AutreLicenceDroit

```

143 Deux documents de travail sont produits :
 — un fichier csv contenant une ligne par candidat·e, et comme colonne les noms, prénoms et numéro de candidat·es, ainsi qu'une colonne « Rang » ;

- un fichier PDF contenant, dans le même ordre, l'intégralité du dossier (formulaire et pièces) fournis par les candidat-es.

⟨PRODUCTIONAUTRELICENCEDROIT⟩ ≡ OCaml

```

1 let autres_licence_droit =
2   List.filter (λx → tri_dossier (snd x) = AutreLicenceDroit)
3     candidats
4 let () = ecriture_csv_travail "autres_licence_droit.csv"
5         ["Nom"; "Prénom"; "Numéro"; "Rang"]
6         (List.map
7           (λc → match c with
8             numero, {nom; prenom} →
9               ↪ [nom; prenom; numero])
10          autres_licence_droit)

```

- 144 Sur la base du rapport mentionné au § 142, la commission classe les candidatures relevant de ce groupe en attribuant un rang de classement à certaines d'entre elles. Plus précisément, pour chaque candidature, elle remplit soit « Non » (si elle ne souhaite pas la classer), soit le numéro de candidat-e de la personne immédiatement au-dessous (si elle souhaite l'interclasser dans la liste produite), soit un rang (si elle souhaite la classer à la fin de la liste). Elle le fait en produisant un document de travail dans lequel elle a rempli la colonne « Rang ».

⟨LECTUREAUTRELICENCEDROIT⟩ ≡ OCaml

```

1 let resultat_autres_licence_droit_inter,
2   resultat_autres_licence_droit_apres,
3   resultat_autres_licence_droit_non =
4   lecture_liste_complementaire
5     "autres_licence_droit.csv" autres_licence_droit

```

7 Cas des bachelors universitaires de technologie carrières juridiques

- 145 Un dispositif dit « passerelle » est mis en place pour les candidatures déposées par des personnes ayant validé au moins les cinq premiers semestres d'un bachelor universitaire de technologie mention Carrières juridiques. Compte tenu des spécificités de ce parcours de formation, et afin d'assurer la prise en compte de profils distincts en rapport avec l'objet du master, ainsi que d'offrir des possibilités de poursuite d'études en master à des candidatures présentant un bon niveau académique issues de ce parcours, une place est réservée à ce groupe de candidatures, sans préjudice de la possibilité d'en admettre un nombre plus élevé. La commission nomme en son sein un-e rapporteur-ice chargé-e d'instruire les dossiers en question.

⟨GROUPESFORMULAIRE 28⟩ + ≡ OCaml

```

1 | BUTCarriereJuridiques

```

- 146 Deux documents de travail sont produits:
- un fichier csv contenant une ligne par candidat-e, et comme colonne les noms, prénoms et numéro de candidat-es, ainsi qu'une colonne « Rang »;

— un fichier PDF contenant, dans le même ordre, l'intégralité du dossier (formulaire et pièces) fournis par les candidat-es.

```

⟨PRODUCTIONBUT⟩ ≡ OCaml
1 let but =
2   List.filter (λx → tri_dossier (snd x) = BUTCarriereJuridiques)
3     candidats
4 let () = ecriture_csv_travail "but.csv"
5     ["Nom"; "Prénom"; "Numéro"; "Rang"]
6     (List.map
7       (λc → match c with
8         numero, {nom = nom; prenom} →
9           ↪ [nom; prenom; numero])
      but)

```

- 147 Sur la base du rapport mentionné au § 145, la commission classe les candidatures relevant de ce groupe en attribuant un rang de classement à certaines d'entre elles. Plus précisément, pour chaque candidature, elle remplit soit « Non » (si elle ne souhaite pas la classer), soit le numéro de candidat-e de la personne immédiatement au-dessous (si elle souhaite l'interclasser dans la liste produite), soit un rang (si elle souhaite la classer à la fin de la liste). Elle le fait en produisant un document de travail dans lequel elle a rempli la colonne « Rang ».

```

⟨LECTUREBUT⟩ ≡ OCaml
1 let resultat_but_inter,
2   resultat_but_apres,
3   resultat_but_non =
4   lecture_liste_complementaire "but.csv" but

```

8 Autres cas

- 148 Toutes les autres candidatures sont regroupées en un cas Autre. Ce cas concerne, de fait, majoritairement des étudiant-es n'ayant pas de licence de droit, ou ayant commencé leur licence de droit dans une université publique française par équivalence (d'une autre discipline, ou un autre type d'établissements). Afin de tenir compte des spécificités de ces situations et d'assurer une évaluation adaptée, la commission désigne en son sein un-e rapporteur-ice chargé-e d'instruire les dossiers en question.

```

⟨GROUPESFORMULAIRE 28⟩ + ≡ OCaml
1 | Autre

```

- 149 Deux documents de travail sont produits :
- un fichier CSV contenant une ligne par candidat·e, et comme colonne les noms, prénoms et numéro de candidat·es, ainsi qu'une colonne « Rang » ;
 - un fichier PDF contenant, dans le même ordre, l'intégralité du dossier (formulaire et pièces) fournis par les candidat·es.

```

⟨PRODUCTIONAUTRE⟩ ≡ OCaml
1 let autres =
2   List.filter (λx → tri_dossier (snd x) = Autre)
3     candidats
4 let () = ecriture_csv_travail "autres.csv"
5     ["Nom"; "Prénom"; "Numéro"; "Rang"]
6     (List.map
7       (λc → match c with
8         numero, {nom; prenom}
9         → [nom; prenom; numero])
10      autres)

```

- 150 Sur la base du rapport mentionné au § 148, la commission classe les candidatures relevant de ce groupe en attribuant un rang de classement à certaines d'entre elles. Plus précisément, pour chaque candidature, elle remplit soit « Non » (si elle ne souhaite pas la classer), soit le numéro de candidat·e de la personne immédiatement au-dessous (si elle souhaite l'interclasser dans la liste produite), soit un rang (si elle souhaite la classer à la fin de la liste). Elle le fait en produisant un document de travail dans lequel elle a rempli la colonne « Rang ».

```

⟨LECTUREAUTRE⟩ ≡ OCaml
1 let resultat_autres_inter,
2   resultat_autres_apres,
3   resultat_autres_non =
4   lecture_liste_complementaire "autres.csv" autres

```

9 Interclassement

- 151 La commission procède à l'interclassement des listes :
- du cas nominal
 - du cas nominal avec saisie défaillante
 - du cas des autres licences de droit
 - des autres cas.
 - des bachelors universitaires de technologie carrières juridiques
- Pour cela, sont produits deux documents de travail pour chacune de ces listes :
- un fichier CSV à quatre colonnes : nom, prénom, numéro et « position », classé dans l'ordre déterminé par la commission ;
 - un fichier PDF contenant l'intégralité des pièces de la liste, dans le même ordre,
- sauf pour la liste du cas nominal, pour laquelle on ne produit que le fichier PDF, dans l'ordre décroissant de score.

- 152 La commission remplit chacun des fichier csv en indiquant, pour chaque candidat·e, le numéro de la candidature immédiatement en dessous duquel la commission souhaite le ou la placer. Elle examine les listes dans l'ordre ci-dessus.

⟨FONCTIONS_CALCUL_45⟩ + ≡ OCaml

```

1 let rec ajouter liste clef clef' principal =
2   match principal with
3     [] → failwith "Mauvais numéro de candidat"
4     | (c,x)::q → if c = clef'
5                   then (clef,List.assoc clef liste)
6                       ::principal
7                   else (c,x)::(ajouter liste clef clef' q)

```

- 153 ⟨FONCTIONS_CALCUL_45⟩ + ≡ OCaml

```

1 let rec interclassement liste principal inter apres =
2   match inter with
3     [] → principal@apres
4     | (clef,avant)::q →
5       interclassement liste
6       (ajouter liste clef avant principal)
7       q apres

```

- 154 ⟨INTERCLASSEMENT⟩ ≡ OCaml

```

1 let candidats_tri =
2   interclassement but
3     (interclassement autres
4       (interclassement autres_licence_droit
5         (interclassement nominaux_saisie_defaillante
6           candidats_tri
7             ⇨ resultat_nominaux_saisie_defaillante_inter
8               resultat_nominaux_saisie_defaillante_apres)
9           resultat_autres_licence_droit_inter
10          resultat_autres_licence_droit_apres)
11        resultat_autres_inter
12        resultat_autres_apres)
13 let candidats_non =
14   resultat_autres_non @
15   resultat_autres_licence_droit_non @
16   resultat_nominaux_saisie_defaillante_non @
17   resultat_but_non @ candidats_sous_seuil

```

- 155 La dernière fonction produit, à partir des listes de rejet et de la liste ci-dessus, un fichier csv à 7 colonnes: Formation; Identifiant candidat; Nom candidat; Prénom candidat; Classement; Motif de refus; Justification dossier invalide. La cinquième colonne est remplie par le rang de classement ou "NC". La sixième colonne contient, si la cinquième contient "NC", NIVEAU INSUFFISANT, et sinon est laissée vide. La commission ne jugeant pas de la recevabilité administrative d'un dossier, elle ne remplit pas la dernière colonne.

⟨SORTIE⟩ ≡ OCaml

```

1 let resultat =
2   ((List.mapi
3     (λr →
4       λx →
5         match x with
6           numero,c
7             → (numero,(c,Some(r+1))))
8     candidats_tri)
9   @(List.map
10    (λx →
11      match x with
12        numero,c
13          → (numero,(c,None)))
14    candidats_non))

```

Étendu en §156, 157, 158.

156 ⟨SORTIE 155⟩ + ≡ OCaml

```

1 let rec ordre_resultat resultat candidats =
2   match candidats with
3     (clef,c)::q → (clef,List.assoc clef resultat)::(ordre_resultat
4       ↪ resultat q)
5   | [] → []

```

157 ⟨SORTIE 155⟩ + ≡ OCaml

```

1 let affiche_resultat r =
2   match r with
3     numero,({nom;prenom},Some(i))
4     → [ ⟨NUMÉROFORMATION 159⟩ ;
5         numero;nom;prenom;string_of_int i]
6   | numero,({nom;prenom},None)
7     → [ ⟨NUMÉROFORMATION 159⟩ ;
8         numero;nom;prenom;"NC";"NIVEAU INSUFFISANT"]

```

158 MonMaster demandant semble-t-il un fichier classé par ordre alphabétique des candidat·es, on le trie.

⟨SORTIE 155⟩ + ≡ OCaml

```

1 let () = ecriture_csv_travail "sortie.csv"
2   ["Formation"; "Identifiant candidat"; "Nom candidat";
3    "Prénom candidat"; "Classement"; "Motif de refus";
4    "Justification dossier invalide"]
5   (List.map affiche_resultat
6     (List.sort
7       (λ(a,_) → λ(b,_) → String.compare a b)
8       resultat))
9   let () = assert (List.length candidats = List.length resultat)

```

159 En 2026 → 2027, la formation avait le numéro suivant.

```
⟨NUMÉROFORMATION⟩ ≡
1 "1501017R54PT"
```

160 Le fichier ainsi produit n'est pas littéralement le fichier téléversé sur la plateforme MonMaster. En effet, celui-ci doit être au format *Office Open XML* (dit XLSX) et contenir un deuxième onglet et un formatage particulier. Le président de la commission d'examen des candidatures est chargé de produire ce fichier et le téléverser.

10 Information des candidat·es

10.1 Information individuelle

161 Une notice d'information individuelle est donnée à chaque candidat·e sur le traitement de sa candidature. Elle contient le plus grand nombre d'informations qui peut être donnée sur le traitement de sa situation individuelle sans donner d'informations sur les autres candidat·es.

162 Le programme contient une fonction pour écrire des fiches de vérification des informations utilisées. Cette fonction est donnée en Annexe, § 278. Si la candidature relevait du groupe nominal ou quasi-nominal, les différents scores calculés sont donnés. Si la candidature était dans un autre cas, son classement ou son rejet est justifié.

```
⟨FONCTIONS EXPOSÉES LECTURE ÉCRITURE 21⟩ + ≡ OCaml
1 val fiche_information : string × (dossier_formulaire × int option)
2   → (string × (dossier_formulaire × ScoreCandidat.t × float)
3     ) list
4   → (dossier_formulaire → groupes_formulaire)
5   → (int array × int array × int array
6     × int array × int array × int array)
7   → (dossier_formulaire → ScoreSemestres_in.t)
8   → unit
```

163 Pour chaque candidature, une fiche décrivant.

```
⟨PRODUCTION INFORMATION⟩ ≡ OCaml
1 let (_, h_baccalaureat, h_autres_diplomes, h_upec, h_appreciation,
2   ↪ h_total)
3   = histogrammes candidats_score_tri
4 let h_semestres = histogramme_complet_semestre nominaux
5 let () = List.iter
6   (λx → fiche_information x
7     candidats_score_tri
8     Calculdossiers.tri_dossier
9     (h_semestres, h_baccalaureat,
10    ↪ h_autres_diplomes,
11    h_upec, h_appreciation, h_total)
12    notes_licence_of_candidat)
```

11 resultat

Étendu en §169.

- 164 Cette fiche est ensuite compilée au format PDF et chiffrée. Pour chaque candidat-e, sa fiche peut être déchiffrée avec comme mot de passe le numéro de candidat-e attribué par la plateforme MonMaster, utilisé comme identifiant individuel non aisément devinable.

```
chiffrement-pdf-information.fish ≡ fish
1 cd "$RepertoireTravailSorties"mail_information_candidats
2 for i in (ls *.tex)
3     lua $\text{latex}$  $i > /dev/null
4     echo $i
5 end
6 for i in (ls *.pdf)
7     mkdir (basename $i .pdf);
8     qpdf --encrypt (basename $i .pdf) (basename $i .pdf) 256 \
9         --accessibility=y \
10        --print=full \
11        --modify=none \
12        -- $i (basename $i .pdf)/dossier.pdf
13 end
```

- 165 La fiche est envoyée aux candidat-es avec un courriel d’accompagnement, contenant un lien vers un formulaire permettant aux candidat-es de signaler des incohérences. Deux courriels existent, l’un pour les candidatures reçues, l’autre pour les candidatures refusées.

- 166 Le courriel pour les candidatures reçues est enregistré dans un fichier texte mail-information-recu.txt.

mail-information-recu.txt ≡

Madame, Monsieur,

Vous avez candidaté au parcours Informatique et droit du Master Droit du numérique de l’Université Paris est Créteil, et nous vous remercions vivement de votre intérêt pour cette formation.

La commission d’examen des candidatures a retenu votre candidature et l’a classée, comme vous avez pu en avoir la notification sur la plateforme MonMaster. Nous nous réjouissons d’avance de vous compter parmi nous à la rentrée, si les conditions s’y prêtent.

Nous nous permettons de vous adresser ce message pour aborder deux points.

- 1° *Pour que votre admission soit définitive, il faut non seulement que votre rang soit au-dessus de la donnée d’appel, mais aussi que vous trouviez un contrat d’alternance. Nous vous invitons à en chercher le plus tôt possible.*

Nous organisons une réunion d’information le lundi 11 mai 2026 à 19h, sur le lien suivant

- 2° *Dans le cadre de la procédure d’examen des candidatures à ce Master, une commission composée de quatre enseignant-chercheur-es a été constituée. Cette commission examine, évalue*

et classe les candidatures en combinant des opérations automatisées, d'une part, et des interventions humaines, d'autre part, selon un processus précis. Un important travail a été mené pour que ce processus soit le plus objectif et le plus transparent possible, ayant bien conscience qu'aujourd'hui, les sélections en Master dans les universités françaises demeurent trop souvent obscures. Ce processus a été formalisé en un programme, disponible à l'adresse suivante: <https://lacl.fr/~lpellissier/master/admission.html> accompagné d'un guide d'explication. Ce programme, ainsi que son guide d'explication sont pensés pour donner un haut niveau d'information sur la manière dont votre candidature a été traitée. Nous vous invitons à regarder ces documents - qui sont caractéristiques des approches développées au sein du master Droit du numérique, au croisement du droit et de l'informatique - et le cas échéant, à nous faire part de vos remarques.

Nous joignons à ce courriel votre notice individuelle d'information. Elle est une représentation de la manière dont le programme d'examen a traité votre candidature spécifiquement, afin de vous donner l'information la plus complète possible sans donner d'information sur les autres candidatures. Cette fiche prend la forme d'un fichier PDF chiffré. Le mot de passe pour y accéder est votre numéro de candidat sur la plateforme MonMaster (ce mot de passe est de la forme CAND suivi de 8 caractères alphanumériques).

Restant à votre disposition pour toute question, nous vous prions de croire en nos meilleures salutations,

- 167 Le courriel pour les candidatures reçues est enregistré dans un fichier texte mail-information-refus.txt.

mail-information-refus.txt ≡

Madame, Monsieur,

Vous avez candidaté au parcours Informatique et droit du Master Droit du numérique de l'Université Paris est Créteil, et nous vous remercions vivement de votre intérêt pour cette formation.

La commission d'examen des candidatures n'a pas retenu votre candidature et ne l'a pas classée, comme vous avez pu en avoir la notification sur la plateforme MonMaster.

Nous nous permettons toutefois de vous adresser ce message afin que vous puissiez comprendre cette décision.

Dans le cadre de la procédure d'examen des candidatures à ce Master, une commission composée de quatre enseignant·es-chercheur·es a été constituée. Cette commission a examiné, évalué et classé les candidatures en combinant des opérations automatisées, d'une part, et des interventions humaines, d'autre part, selon un processus précis. Un important travail a été mené pour que ce processus soit le plus objectif et le plus transparent possible, ayant bien conscience qu'aujourd'hui, les sélections en Master dans les universités françaises demeurent trop souvent obscures.

Ce processus a été formalisé en un programme, disponible à l'adresse suivante: <https://lacl.fr/~lpellissier/master/admission.html> accompagné d'un guide d'explication. Ce programme, ainsi que son guide d'explication sont pensés pour donner un haut niveau d'information sur la manière dont votre candidature a été traitée. Nous vous invitons à regarder ces documents — qui sont caractéristiques des approches développées au sein du master Droit du numérique, au croisement du droit et de l'informatique — et le cas échéant, à nous faire part de vos remarques.

Nous joignons à ce courriel votre notice individuelle d'information. Elle est une représentation de la manière dont le programme d'examen a traité votre candidature spécifiquement, afin de vous donner l'information la plus complète possible sans donner d'information sur les autres candidatures. Cette fiche prend la forme d'un fichier PDF chiffré. Le mot de passe pour y accéder est votre numéro de candidat sur la plateforme Mon/Master (ce mot de passe est de la forme CAND suivi de 8 caractères alphanumériques).

Restant à votre disposition pour toute question, nous vous prions de croire en nos meilleures salutations,

168 Ce courriel est envoyé par l'opérateur du traitement à chaque candidat·e.

```

courriel-information.el ≡ emacs-lisp
1  <SIGNATUREMAIL 43>
2  (let (message-mail-user-agent)
3    (dolist
4      <LECTUREFICHIERCANDIDATSINFORMATION 170>
5        (let ((form-letter
6              (lambda ()
7                (save-excursion
8                  (goto-char (- (point-max) 45))
9                  <PIÈCEJOINTEINFORMATION 171>
10                 <CONTENUMAILINFORMATION 172>
11                 )))
12          (unwind-protect
13            (progn
14              (add-hook 'message-setup-hook form-letter)
15              <TITREMAILINFORMATION 173>
16              )
17            (remove-hook 'message-setup-hook form-letter))))))

```

169 On crée un fichier csv des candidat·es à qui envoyer le courriel, séparé par des points virgules, dont la première colonne est le numéro de candidat·e et la deuxième l'adresse email.

```

<PRODUCTIONINFORMATION 163> + ≡ OCaml
1  let () = ecriture_csv_travail
2    "mail_information_candidats/candidats.csv"
3    []
4    (List.map
5      (λc → match c with
6        numero, {adresse_email}

```

```

7           → [numero;adresse_email;
8             (match (List.assoc numero resultat)
9               with (_,Some(i)) → string_of_int i
10              | (_,None) → "")])
11         candidats)

```

170 Ce fichier csv est ensuite lu pour l'envoi.

```

<LECTUREFICHIERCANDIDATSINFORMATION> ≡ emacs-lisp
1 (candidat (cdr
2   (nbutlast
3     (with-temp-buffer
4       (insert-file-contents
5         (concat <REPertoireTRAVAILSORTIES 188>
6           "mail_information_candidats/"
7           "candidats.csv"))
8       (split-string (buffer-string) "\n"))))
9 (cl-destructuring-bind (numero email rang)
10   (split-string candidat ";"))

```

```

171 <PIECEJOINTEINFORMATION> ≡ emacs-lisp
1 (insert "<#part type=\"application/pdf\" filename=\"\"
2   <REPertoireTRAVAILSORTIES 188> \"mail_information_candidats/\"
3   numero \"/dossier.pdf\" disposition=attachement><#/part>\n")

```

```

172 <CONTENUMAILINFORMATION> ≡ emacs-lisp
1 (insert-file-contents (concat <REPertoireTRAVAILSORTIES 188>
2   (if (> (string-to-number rang) 0)
3     "mail-information-recu.txt"
4     "mail-information-refus.txt")))

```

```

173 <TITREMAILINFORMATION> ≡ emacs-lisp
1 (message-mail email
2   "[Master droit du numérique UPEC] Information sur
   ↪ votre candidature")

```

10.2 Rapport du jury

174 Conformément au § 7, des éléments statistiques sont produits afin de permettre l'élaboration, au plus tard le 28 octobre 2026, du rapport de la commission prévu au troisième alinéa de l'article D. 612-36-2-1 du code de l'éducation. Les opérations correspondantes relèvent de l'exécution du programme, ainsi que le prévoit le § 179. Une version ultérieure du programme, publiée dans les mêmes conditions que la présente version, a vocation à intégrer explicitement les opérations permettant la production automatisée de ces éléments statistiques.

II Divers

- 175 Six mois après l'information des candidat·es, l'opérateur·ice du traitement détruit l'intégralité des entrées et des documents de travail, sans préjudice des obligations légales d'archivage.
- 176 La commission conçoit le présent programme comme pouvant, en tant que tel, faire l'objet de recours, y compris en justice.
- 177 Le programme est distribué sous le contrat de licence de logiciel libre CeCILL susvisé.
- 178 Sous le contrôle des autres membres de la commission d'examen des candidatures, son ou sa président·e et l'opérateur·ice du traitement sont chargé·es de l'exécution du présent programme.
- 179 L'exécution du présent programme a commencé le 2 avril 2026 par la production des premiers documents de travail (§§ 130, 138, 143 et 146), elle s'est poursuivie le 14 avril par l'envoi des courriels de vérification aux candidat·es (§ 37), puis le 16 avril par la fixation du seuil mentionné au § 59, le 27 avril par le téléversement de la sortie principale (§ 160), et le 7 mai 2026 par l'envoi des notices individuelles d'information aux candidat·es (§ 168).
Elle se poursuivra ultérieurement par l'élaboration du rapport du jury mentionné au § 7, et la destruction et l'archivage mentionnés au § 175.



Fait à Créteil, le 7 mai 2026,

La commission d'examen des candidatures :

P^R PIERRE VALARCHER
président

P^{RE} LAURIE MARGUET

LUC PELLISSIER

P^R NOÉ WAGENER

1. *Le présent programme est susceptible de faire l'objet de recours en tant qu'il fixe les règles et modalités de l'ensemble des opérations automatisées et non automatisées permettant l'examen et l'évaluation des candidatures au parcours Informatique et Droit du Master Droit du numérique de l'université Paris est Créteil, en vue de l'établissement de leur classement. Dans un délai de deux mois à compter de l'accomplissement des formalités adéquates de publicité de ce programme, les recours suivants peuvent être introduits à l'encontre de ce programme, conformément aux dispositions de l'article R. 421-1 et suivants du code de justice administrative et du livre IV du code des relations entre le public et l'administration :*
 - *un recours gracieux adressé à la présidente de l'université Paris est Créteil — 61, avenue du Général de Gaulle — 94010 Créteil Cedex*;
 - *un recours contentieux en saisissant le tribunal administratif de Melun — 43, rue du Général de Gaulle — Case postale n° 8630 — 77008 Melun Cedex ou par l'application <https://www.telerecours.fr>.*
2. *Les décisions individuelles relatives aux candidatures (classement ou refus), produites à l'issue de l'ensemble des opérations automatisées et non automatisées prévues par le présent programme, peuvent, en outre, faire l'objet de voies de recours distinctes. Dans un délai de deux mois à compter de leur notification, ces décisions peuvent faire l'objet :*
 - *d'un recours gracieux adressé à la présidente de l'université Paris est Créteil, dans les mêmes conditions que ci-dessus ;*
 - *d'un recours contentieux en saisissant le tribunal administratif de Melun, dans les mêmes conditions que ci-dessus.*
3. *Les personnes dont des données à caractère personnel sont traitées dans le cadre du présent programme peuvent, par ailleurs exercer leurs droits d'accès, de rectification et de limitation sur ces données, conformément aux dispositions des articles 15, 16 et 18 du règlement UE 2016/679 du Parlement européen et du Conseil du 27 avril 2016 relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données, et abrogeant la directive 95/46/CE, ainsi que leur droit d'opposition, conformément aux dispositions de l'article 21 du même règlement.*
Ces demandes doivent être adressées à la déléguée à la protection des données de l'université Paris est Créteil, soit via l'adresse électronique : dpo@u-pec.fr, soit par courrier postal à l'attention de cette déléguée à la protection des données, à la même adresse que le courrier adressé à la présidente mentionné ci-dessus.
Si, après avoir introduit ces demandes, aucune réponse n'est apportée dans le délai d'un mois, éventuellement prolongé dans les conditions prévues par les dispositions de l'article 12 § 3 du règlement ou si cette suite n'est pas considérée comme satisfaisante, une réclamation peut être adressée auprès de la CNIL à l'adresse suivante : 3, place du Fontenoy — TSA 80715 — 75334 Paris Cedex 07.
4. *Les différents éléments produits dans le cadre du programme, mentionnés au § 4, peuvent être communiqués sur le fondement des dispositions des articles L. 311-1 et suivants du code des relations entre le public et l'administration et des articles L. 213-1 et suivants du code du patrimoine, sous réserve des dispositions des articles L. 311-5 et L. 311-6 du code des relations entre le public et l'administration, et des délais prévus aux articles L. 213-2 et L. 213-3 du code du patrimoine.*
Si la communication est refusée par l'autorité à laquelle la demande a été faite, ou si celle-ci n'a pas répondu dans le délai d'un mois, la commission d'accès aux documents administratifs peut être saisie dans un délai de deux mois à compter de la notification du refus ou de l'intervention du refus tacite, conformément aux dispositions des articles R. 343-1 et suivants du code des relations entre le public et l'administration. Ce recours devant la CADA constitue un préalable obligatoire à tout recours contentieux, conformément aux dispositions de l'article L. 342-1 du même code. Il peut être formé :
 - *sur papier libre, par lettre simple — 20, avenue de Ségur, 75007 Paris ;*
 - *par voie électronique à l'adresse cada@cada.fr ;*
 - *ou via le formulaire en ligne <https://www.cada.fr/formulaire-de-saisine>.*
5. *Indépendamment de ces voies de recours, il est possible de prendre contact avec la commission d'examen des candidatures, afin d'échanger à propos du programme et de sa mise en oeuvre. Ces échanges, qui ne constituent pas des recours administratifs mais un simple canal de communication, n'interrompent ni ne suspendent les délais de recours. Le courrier doit être adressé au président de la commission d'examen des candidatures au parcours Informatique et Droit du Master Droit du numérique — UFR de droit — 83-85, avenue du Général de Gaulle — 94010 Créteil Cedex.*

Table des matières

1	Principes	4
2	Architecture du programme	7
3	Cas nominal	9
3.1	Vérification des données du formulaire	10
3.1.1	Information	10
3.1.2	Correction	15
3.2	Calcul sur les notes de licence	15
3.2.1	Coefficients et paramètres	17
3.3	Traitement des candidatures dont le score est inférieur à un seuil	18
3.4	Appréciation des candidatures dont le score est supérieur du seuil ou qui ont été rattrapées	20
3.5	Calcul sur l'intégralité du dossier	23
3.6	Baccalauréat	24
3.6.1	Baccalauréat général	24
3.6.2	Baccalauréat technologique	27
3.6.3	Baccalauréat professionnel	28
3.6.4	Coefficients des baccalauréats	29
3.6.5	Équivalences et baccalauréats étrangers	30
3.7	Master et licence	30
3.8	Établissement d'origine	31
3.9	Somme des scores	32
4	Cas quasi-nominal	35
5	Cas nominal avec saisie défailante	37
6	Cas des autres licences de droit	38
7	Cas des bachelors universitaires de technologie carrières juridiques	39
8	Autres cas	40
9	Interclassement	41
10	Information des candidat-es	44
10.1	Information individuelle	44
10.2	Rapport du jury	48
11	Divers	49
	Annexes	52
A	Disposition des fichiers	52
B	Fonctions opérant sur les fichiers csv	55
C	Lecture du fichier fourni par la plateforme MonMaster	56
C.1	Lecture du formulaire	56
C.2	Construction d'une candidate	72
D	Manipulation des dossiers	80
D.1	Tri des dossiers en groupes	80
D.2	Mise en forme des dossiers (quasi)nominaux	83
D.3	Autres fonctions de lecture	84
E	Fonctions d'écriture de fichiers de travail	84
E.1	Fiches de vérification	84
E.2	Liasses	86
E.3	Fiches d'information finales	91
F	Index	101

*Programme du 7 mai 2026 d'examen et d'évaluation
des candidatures à l'admission en première année du
parcours Informatique & droit pour l'année
universitaire 2026 → 2027*

ANNEXES

A Disposition des fichiers

180 Le présent programme s'enchevêtre en plusieurs fichiers. La présente section les décrit, ainsi que les paramètres des fichiers d'entrée, de travail et de sortie.

181 Un module `CalculScoresCasNominal`, écrit en `Catala` et traduit en `OCaml`, contient les règles de calcul des scores des candidat·es pour lesquels un score est calculé.

`CalculScoresCasNominal.catala_fr` ≡ Catala

```
1 > Module CalculScoresCasNominal
2 <CALCULSCORESCASNOMINAL 29>
```

182 Un module `Lecture_écriture`, écrit en `OCaml`, contient l'intégralité des fonctions de lecture et d'écriture de fichiers.

`lib/lecture_écriture/lecture_écriture.ml` ≡ OCaml

```
1 open CalculScoresCasNominal
2 <TYPESDOSSIERS 20>
3 <FONCTIONSCSV 200>
4 <LECTUREFORMULAIRE 207>
5 <LECTURELISTECOMPLÉMENTAIRE 141>
6 <FICHEVÉRIFICATION 263>
7 <LIASSENOMINAUX 271>
8 <FICHEINFORMATION 278>
```

183 `lib/lecture_écriture/lecture_écriture.mli` ≡ OCaml

```
1 <TYPESDOSSIERS 20>
2 <FONCTIONSÉXPOSÉESLECTUREÉCRITURE 21>
3 <DÉCLARATIONFONCTIONSLECTURE 64>
4 <DÉCLARATIONFONCTIONSÉCRITURE 202>
5 val liasse_nominaux :
6   string
7   → (string × (dossier_formulaire × ScoreCandidat.t × float))
8   ↪ list
9   → (int array × int array × int array × int array × int array
10  ↪ × int array)
11  → unit
```

- 184 Un module `CalculsDossiers`, écrit en OCaml, contient des fonctions qui opèrent sur les dossiers.

```
lib/calculdossiers/calculdossiers.ml ≡ OCaml
1  open CalculScoresCasNominal
2  open Lecture_écriture
3  module CalculScoresCasNominal = CalculScoresCasNominal
4  module Lecture_écriture = Lecture_écriture
5  ⟨FONCTIONS_CALCUL 45⟩
```

- 185 Le fichier principal, écrit en OCaml, contient des fonctions qui opèrent sur les dossiers, et la suite d'instructions de la procédure.

```
bin/main.ml ≡ OCaml
1  open CalculScoresCasNominal
2  open Lecture_écriture
3  open CalculDossiers
4  ⟨PROCÉDURE 25⟩
```

- 186 Pour son exécution, on télécharge les pièces jointes dans un répertoire.

```
⟨REPERTOIRE_ENTREES⟩ ≡ ...
```

- 187 Et on suppose que le fichier des formulaires `y` est présent et s'intitule

```
⟨FICHIER_FORMULAIRE⟩ ≡ ...
```

- 188 Les fichiers de travail produits informatiquement sont placés dans un répertoire

```
⟨REPERTOIRE_TRAVAIL_SORTIES⟩ ≡ ...
```

- 189 Les fichiers de travail remplis manuellement par la commission sont placés dans un répertoire

```
⟨REPERTOIRE_TRAVAIL_ENTREES⟩ ≡ ...
```

- 190 Fichier `clerk.toml`

```
clerk.toml ≡
1  [project]
2  include_dirs = [ "." ]
3  build_dir    = "_build"
4  target_dir   = "_target"
5
6  [[target]]
7  name        = "monmaster"
8  modules     = [ "CalculScoresCasNominal" ]
9  tests       = [ "tests.catala_fr" ]
10 backends    = [ "ocaml" ]
```

- 191 Fichier `lib/CalculScoresCasNominal/dune`

```
lib/CalculScoresCasNominal/dune ≡
1  (library
2   (name CalculScoresCasNominal)
3   (libraries zarith))
```

192 Fichier lib/lecture_ecriture/dune

```
lib/lecture_ecriture/dune ≡
1 (library
2   (name lecture_ecriture)
3   (libraries re csv (re_export CalculScoresCasNominal)))
```

193 Fichier lib/calculdossiers/dune

```
lib/calculdossiers/dune ≡
1 (library
2   (name calculdossiers)
3   (libraries (re_export lecture_ecriture)
4             (re_export CalculScoresCasNominal)))
```

194 Fichier bin/dune

```
bin/dune ≡
1 (executable
2   (public_name monmaster)
3   (name main)
4   (libraries calculdossiers lecture_ecriture
5             CalculScoresCasNominal zarith))
```

195 La bibliothèque mathématique a été écrite avec l'aide de Vincent Botbol.

196 Fichier math.catala_fr

```
1 > Module Math externe
2
3 ````catala-metadata
4 déclaration pow
5     contenu décimal dépend de
6     base contenu décimal,
7     exposant contenu décimal
8 ````
```

197 Fichier math.mli

```
1 open Catala_runtime
2
3 val pow : decimal → decimal → decimal
```

198 Fichier math.ml

```
1 open Catala_runtime
2
3 let pow (q : decimal) (exp : decimal) : decimal =
4   let qf : float = Q.to_float q in
5   let expf : float = Q.to_float exp in
6   Q.of_float (qf ** expf)
7
8 let () =
9   Catala_runtime.register_module "Math"
```

```

10     [
11         "pow", StdLib.Obj.repr pow;
12     ]
13     "*external*"

```

B Fonctions opérant sur les fichiers csv

199 Les fichiers fournis et demandés par la plateforme *MonMaster* sont des tableaux. On utilise donc le format csv comme format d'échange et de travail. On utilise la bibliothèque csv d'*OCaml* de manière minimale, et au travers de principalement trois fonctions.

200 La fonction

```

1  val chargement_csv : string → string list list

```

prend en entrée un nom de fichier csv et en produit une représentation de toutes les lignes sauf la première sous forme de liste de listes de chaînes de caractères, en utilisant la bibliothèque csv.

⟨FONCTIONS_CSV⟩ ≡ OCaml

```

1  let chargement_csv (nom_fichier : string) : string list list =
2      match (try (Csv.load ~separator:';' nom_fichier) with
3              Sys_error _ →
4                  ⟨FICHIER_ABSENT_206⟩
5              ) with
6      | t::formulaire → formulaire
7      | _ → failwith "Pas de ligne de titre"

```

Étendu en §201, 203.

201 ⟨FONCTIONS_CSV_200⟩ +≡ OCaml

```

1  let lecture_csv_travail (nom : string) : string list list =
2      chargement_csv ( ⟨REPERTOIRE_TRAVAIL_ENTRÉES_189⟩ ^ nom)

```

202 ⟨DÉCLARATION_FONCTIONS_ÉCRITURE⟩ ≡ OCaml

```

1  val ecriture_csv_travail : string → string list
2                               → string list list
3                               → unit

```

203 ⟨FONCTIONS_CSV_200⟩ +≡ OCaml

```

1  let ecriture_csv_travail (nom : string)
2      (intitules_colonnes : string list)
3      (colonnes : string list list) : unit =
4      Csv.save ~separator:';' ( ⟨REPERTOIRE_TRAVAIL_SORTIES_188⟩ ^ nom)
5      (intitules_colonnes::colonnes)

```

C Lecture du fichier fourni par la plateforme MonMaster

204 Les données fournies par la plateforme MonMaster se présentent sous la forme d'un fichier PDF pour chaque candidature contenant les réponses au formulaire et les pièces, dont le nom est la référence de la formation¹, suivie d'un trait d'union et du numéro de candidat² et l'extension .pdf (par exemple: 1501017R54PT-CANDABCD1234.pdf), et d'un tableau contenant les réponses au formulaire dont chaque ligne est un-e candidat-e, et les colonnes sont décrites dans les paragraphes suivants. Le format de ce fichier n'est pas documenté, et ce qui suit n'est qu'une description empirique.

205 La première ligne du fichier contient les titres des colonnes; il faut donc l'ignorer.

206 \langle FICHIERABSENT $\rangle \equiv$ OCaml

```

1 failwith ("///!\ Le fichier " ^ nom_fichier ^
2           " doit être créé par la commission à l'emplacement" ^
3           " de travail //!\")

```

C.1 Lecture du formulaire

207 La fonction

```
1 val formulaire : string → string list list
```

vérifie de plus que les cases du formulaire utilisées par le programme sont bien au format attendu.

\langle LECTUREFORMULAIRE $\rangle \equiv$ OCaml

```

1 let formulaire (nom_fichier : string) : string list list =
2   let formulaire = chargement_csv nom_fichier in
3    $\langle$ VÉRIFICATIONS 215 $\rangle$ 
4   formulaire

```

Étendu en §208.

208 \langle LECTUREFORMULAIRE 207 $\rangle + \equiv$ OCaml

```

1  $\langle$ CONSTRUCTIONBACCALAURÉAT 242 $\rangle$ 
2  $\langle$ CONSTRUCTIONLICENCE 233 $\rangle$ 
3  $\langle$ CANDIDAT 209 $\rangle$ 

```

209 La fonction

```
1 val construction_candidat : string list → string ×
   ↪ dossier_formulaire
```

construit un candidat à partir d'une ligne du fichier du formulaire.

\langle CANDIDAT $\rangle \equiv$ OCaml

1. Ici 1501017R54PT.

2. Huit caractères alphanumériques suivant le préfixe CAND.

```

1 let construction_candidat (ligne : string list)
2   : string × dossier_formulaire =
3   match ligne with
4     ⟨LIGNECANDIDAT 211⟩
5     ::q →
6       ⟨CONSTRUCTIONCANDIDAT 231⟩
7   | _ →
8     ⟨LIGNEINVALIDE 202⟩

```

210 $\langle \text{CANDIDAT } 209 \rangle + \equiv$ OCaml

```

1 let construction_candidats (nom_fichier : string)
2   : (string × dossier_formulaire) list =
3   List.map construction_candidat
4     (formulaire nom_fichier)

```

211 La première colonne contient uniquement la référence de la formation (dans notre cas 1501017R54PT). Elle peut être oubliée.

1. Référence

$\langle \text{LIGNECANDIDAT} \rangle \equiv$ OCaml

1 -

Étendu en §212, 213, 214, 217, 218, 220, 221, 225, 226, 228, 230.

212 Le premier groupe de colonnes est une suite d'informations personnelles sur le candidat·e. Parmi celles-ci, seuls le nom, le prénom, le numéro de candidat (pour récupérer les fichiers déposés) et l'adresse email (pour informer le candidat·e) sont enregistrés.

2. Statut de la candidature
3. Statut précédent de la candidature
4. Date du dernier changement de statut
5. Nom de naissance
6. Prénom
7. Numéro de candidat
8. Démissionnaire
9. Validation administrative
10. Adresse e-mail
11. Numéro de téléphone principal
12. Numéro de téléphone secondaire
13. Civilité
14. INE/INA/BEA saisi par le candidat
15. INE maître (INES)
16. Date de naissance
17. Nationalité
18. Statut de réfugié
19. Adresse

20. Complément d'adresse

21. Code Postal

22. Ville / Commune

23. Pays de résidence

⟨LIGNECANDIDAT 211⟩ + ≡

OCaml

```

1  ::_
2  ::_
3  ::_
4  ::nom                (* string *)
5  ::prenom            (* string *)
6  ::numero            (* string *)
7  ::_
8  ::_
9  ::adresse_email    (* string *)
10 ::_
11 ::_
12 ::_
13 ::_
14 ::_
15 ::_
16 ::_
17 ::_
18 ::_
19 ::_
20 ::_
21 ::_
22 ::_

```

213 Celui-là est très mystérieux, et vide pour toutes les candidat·es.

24. Missions du poste

⟨LIGNECANDIDAT 211⟩ + ≡

OCaml

```

1  ::_

```

214 Les quatre colonnes suivantes décrivent le baccalauréat. Des informations, nous ne gardons que le type et la série.

25. Année d'obtention du baccalauréat

26. Type de baccalauréat ou titre admis en équivalence

27. Série du baccalauréat

28. Mention obtenue

⟨LIGNECANDIDAT 211⟩ + ≡

OCaml

```

1  ::_
2  ::type_baccalaureat (* type à déterminer *)
3  ::serie_baccalaureat (* type à déterminer *)
4  ::_

```

215 Empiriquement, le type du baccalauréat est soit vide, soit une des chaînes de caractères suivantes :

- AUTRE CAS DE NON BACHELIERS
- CAPACITE DE DROIT
- Général
- PROMOTION SOCIALE
- Professionnel
- TITRE ETRANGER ADMIS EN EQUIVALENCE (dont bac franco-allemand, Abibac, Bachibac et Esabac)
- TITRE FRANCAIS ADMIS EN DISPENSE
- Technologique
- VALIDATION D'ETUDES, D'EXPERIENCES PROFESSIONNELLES, D'ACQUIS PERSONNELS

Ce que l'on peut vérifier par

(VÉRIFICATIONS) \equiv OCaml

```

1 let type_bac_autorises =
2   [""; "AUTRE CAS DE NON BACHELIERS"; "CAPACITE DE DROIT";
   ↪ "Général";
3   "PROMOTION SOCIALE"; "Professionnel";
4   "TITRE ETRANGER ADMIS EN EQUIVALENCE (dont bac franco-allemand,
   ↪ Abibac, Bachibac et Esabac)";
5   "TITRE FRANCAIS ADMIS EN DISPENSE"; "Technologique";
6   "VALIDATION D'ETUDES, D'EXPERIENCES PROFESSIONNELLES, D'ACQUIS
   ↪ PERSONNELS"]
7   in
8   assert (List.fold_left
9     (λacc ->
10      λx ->
11       acc && (List.mem (List.nth x 25) type_bac_autorises))
12     true formulaire);
```

216 Empiriquement, la série du baccalauréat est soit vide, soit une des chaînes de caractères suivantes :

- Bacs professionnels tertiaires
- E
- Economique et social (ES)
- Littéraire (L)
- Sciences et technologies de la santé et du social (ST2S)
- Sciences et technologies de laboratoire (STL)
- Sciences et technologies de l'industrie et du développement durable (STI2D)
- Sciences et technologies du management et de la gestion (STMG)
- Scientifique (S)

soit la concaténation par | d'une des chaînes de caractères suivantes :

- Arts plastiques (ARTS_ARTS PLA)
- Danse (ARTS_DANSE)

- Histoire géographie, géopolitique et sciences politiques (HGGSP)
- Humanités, littérature et philosophie (HLP)
- LLCER AMC – Langues, littératures et cultures étrangères et régionales
Anglais monde contemporain
- LLCER Américain – Langues, littératures et cultures étrangères et
régionales Américain
- LLCER Anglais – Langues, littératures et cultures étrangères et régionales
Anglais
- LLCER Espagnol – Langues, littératures et cultures étrangères et régionales
Espagnol
- Mathématiques (MATHS)
- Numérique et sciences informatiques (NSI)
- Physique–chimie (PH_CH)
- Sciences de la vie et de la Terre (SVT)
- Théâtre (ARTS_THEATRE)
- sciences économiques et sociales (SES)

ce que l'on peut vérifier par :

(VÉRIFICATIONS 215) + ≡

OCaml

```

1 let serie_baccalaureat_autorisees =
2   [""; "Bacs professionnels tertiaires"; "E"; "Economique et social
   ↪ (ES)";
3   "Littéraire (L)"; "Sciences et technologies de la santé et du
   ↪ social (ST2S)";
4   "Sciences et technologies de laboratoire (STL)";
5   "Sciences et technologies de l'industrie et du développement
   ↪ durable (STI2D)";
6   "Sciences et technologies du management et de la gestion (STMG)";
7   "Scientifique (S)"]
8 and specialite_baccalaureat_autorisees =
9   [""; "Arts plastiques (ARTS_ARTS PLA)";
10  "Danse (ARTS_DANSE)";
11  "Histoire géographie, géopolitique et sciences politiques
   ↪ (HGGSP)";
12  "Humanités, littérature et philosophie (HLP)";
13  "LLCER AMC – Langues, littératures et cultures étrangères et
   ↪ régionales Anglais monde contemporain";
14  "LLCER Américain – Langues, littératures et cultures étrangères
   ↪ et régionales Américain";
15  "LLCER Anglais – Langues, littératures et cultures étrangères et
   ↪ régionales Anglais";
16  "LLCER Espagnol – Langues, littératures et cultures étrangères et
   ↪ régionales Espagnol";
17  "Mathématiques (MATHS)";
18  "Numérique et sciences informatiques (NSI)"; "Physique–chimie
   ↪ (PH_CH)";
19  "Sciences de la vie et de la Terre (SVT)";
20  "Théâtre (ARTS_THEATRE)"; "sciences économiques et sociales
   ↪ (SES)"]

```

```

21   in
22   assert (List.fold_left
23     (λacc →
24       λx →
25         let serie_specialite = (List.nth x 26) in
26         match (String.split_on_char '|' serie_specialite) with
27         | t::[] → acc && (List.mem t
28           ↪ serie_baccalaureat_autorisees)
29         | s1::s2::[] → acc && (List.mem s1
30           ↪ specialite_baccalaureat_autorisees)
31           && (List.mem s2
32             ↪ specialite_baccalaureat_autorisees)
33         | _ → failwith "Ni une ni deux spécialités")
34     true formulaire);

```

217 Expériences non-universitaires. Elles ne sont pas prises en compte.

29. Engagements citoyens
30. Sportif haut niveau
31. Artiste confirmé
32. Situation de handicap
33. Césure envisagée ?
34. Entrepreneur
35. Situation actuelle
36. Avez-vous engagé une démarche de financement dans le cadre de la formation continue ?
37. Stages - Nombre
38. Stages : Employeurs - Durée (en semaines)
39. Stages : Durée totale (en semaines)
40. Expériences - Nombre
41. Expériences : Employeurs - Durée (en mois)
42. Expériences : Durée totale (en mois)

(LIGNECANDIDAT 211) + ≡

OCaml

```

1   ::_
2   ::_
3   ::_
4   ::_
5   ::_
6   ::_
7   ::_
8   ::_
9   ::_
10  ::_
11  ::_
12  ::_
13  ::_
14  ::_

```

218 Les colonnes suivantes concernent l'année Bac+3. Les paragraphes suivant (pour les autres années) sont construits à l'identique. Elles sont, en général, composées de plusieurs lignes (séparées par un changement de ligne (U+000A) '\n'), et les différentes lignes se correspondent. La première colonne comprend soit une information indiquant que le reste est vide, soit un type de diplôme et d'année. La deuxième colonne contient l'année calendaire, la troisième et la quatrième la mention et le parcours (le parcours peut, dans certains cas, être vide). Enfin, on extrait aussi les moyennes des deux semestres, et l'établissement d'inscription.

43. Année bac+3
44. Année universitaire / civile (si calendrier austral)
45. Mention / spécialité / description du diplôme / formation dans le cas de diplôme étranger / Autre
46. Parcours
47. Césure ?
48. Pas de moyenne au semestre ?
49. Moyenne au premier semestre
50. Moyenne au second semestre
51. Relevés de notes au premier semestre
52. Relevés de notes au second semestre
53. Relevé(s) de notes traduits ?
54. Justification d'absence des relevés
55. Établissement
56. Aménagement d'études
57. Avez-vous suivi cette formation dans le cadre d'une mobilité internationale ?

⟨LIGNECANDIDAT 211⟩ + ≡

OCaml

```

1  ::inscription_annee_bac_plus_3
2  ::annee_annee_bac_plus_3
3  ::mention_annee_bac_plus_3
4  ::parcours_annee_bac_plus_3
5  ::_
6  ::_
7  ::moyenne_s1_annee_bac_plus_3
8  ::moyenne_s2_annee_bac_plus_3
9  ::_
10 ::_
11 ::_
12 ::_
13 ::etablissement_annee_bac_plus_3
14 ::_
15 ::_

```

219 On vérifie que les éléments de la case 43 sont bien de la forme donnée.

⟨VÉRIFICATIONS 215⟩ + ≡

OCaml

```

1 Let description_annee_l3_autorisee =
2   ["Aucun cursus post-bac ni diplômé déclaré ";
3    "BACHELOR (3ème année)";
4    "BUT3 (3ème année de bachelor universitaire de technologie)";
5    "Bachelor+4 (3ème année)"; "DCG (3ème année)"; "DU Bac+3 (3ème
   ↪ année)";
6    "Diplôme d'Etat Bac+3 (3ème année)"; "Diplôme étranger ";
7    "INGENIEUR (3ème année)"; "L3 (3ème année de licence)";
8    "LP3 (3ème année de licence professionnelle)"; "Non"]

```

220 Année Bac+2

58. Année bac+2
59. Année universitaire / civile (si calendrier austral)
60. Mention / spécialité / description du diplôme / formation dans le cas de diplôme étranger / Autre
61. Parcours
62. Césure ?
63. Pas de moyenne au semestre ?
64. Moyenne au premier semestre
65. Moyenne au second semestre
66. Relevés de notes au premier semestre
67. Relevés de notes au second semestre
68. Relevé(s) de notes traduits ?
69. Justification d'absence des relevés
70. Établissement
71. Aménagement d'études
72. Avez-vous suivi cette formation dans le cadre d'une mobilité internationale ?

(LIGNECANDIDAT 211) + ≡

OCaml

```

1  ::inscription_annee_bac_plus_2
2  ::annee_annee_bac_plus_2
3  ::mention_annee_bac_plus_2
4  ::parcours_annee_bac_plus_2
5  ::_
6  ::_
7  ::moyenne_s1_annee_bac_plus_2
8  ::moyenne_s2_annee_bac_plus_2
9  ::_
10 ::_
11 ::_
12 ::_
13 ::etablissement_annee_bac_plus_2
14 ::_
15 ::_

```

On vérifie que les éléments de la case 58 sont bien de la même forme donnée.

(VÉRIFICATIONS 215) + ≡

OCaml

```

1  and description_annee_l2_autorisee =
2  ["Aucun cursus post-bac ni diplômé déclaré "; "Autre diplôme
3  ↪ français ";
4  "BACHELOR (2ème année)"; "BTS (2ème année)";
5  "BUT2 (2ème année de bachelor universitaire de technologie)";
6  "Bachelor+4 (2ème année)"; "CPGE Bac+2 (2ème année)"; "DCG (2ème
7  ↪ année)";
8  "DEUG (2ème année)"; "DU Bac+2 (2ème année)"; "DU Bac+3 (2ème
9  ↪ année)";
10 "DU Bac+5 (2ème année)"; "DUT (2ème année)";
11 "Diplôme d'Etat Bac+2 (2ème année)"; "Diplôme étranger ";
12 "L2 (2ème année de licence)"; "LP2 (2ème année de licence
13 ↪ professionnelle)";
14 "Non"]

```

221 Année Bac+1

73. Année bac+1
74. Année universitaire / civile (si calendrier austral)
75. Mention / spécialité / description du diplôme / formation dans le cas de diplôme étranger / Autre
76. Parcours
77. Césure ?
78. Pas de moyenne au semestre ?
79. Moyenne au premier semestre
80. Moyenne au second semestre
81. Relevés de notes au premier semestre
82. Relevés de notes au second semestre
83. Relevé(s) de notes traduits ?
84. Justification d'absence des relevés
85. Établissement
86. Aménagement d'études
87. Avez-vous suivi cette formation dans le cadre d'une mobilité internationale ?

⟨LIGNECANDIDAT 211⟩ + ≡

OCaml

```

1  ::inscription_annee_bac_plus_1
2  ::annee_annee_bac_plus_1
3  ::mention_annee_bac_plus_1
4  ::parcours_annee_bac_plus_1
5  ::_
6  ::_
7  ::moyenne_s1_annee_bac_plus_1
8  ::moyenne_s2_annee_bac_plus_1
9  ::_
10 ::_
11 ::_
12 ::_
13 ::etablissement_annee_bac_plus_1
14 ::_
15 ::_

```

222 On vérifie que les éléments de la case 73 sont bien de la même forme donnée.

(VÉRIFICATIONS 215) + ≡ OCaml

```

1 and description_annee_l1_autorisee =
2   ["Aucun cursus post-bac ni diplômé déclaré "; "Autre diplôme
   ↪ français ";
3   "BACHELOR (1ère année)"; "BTS (1ère année)";
4   "BUT1 (1ère année de bachelor universitaire de technologie)";
5   "Bachelor+4 (1ère année)"; "CPGE Bac+2 (1ère année)"; "DCG (1ère
   ↪ année)";
6   "DOCT MED (1ère année)"; "DU Bac+1 (1ère année)"; "DU Bac+2 (1ère
   ↪ année)";
7   "DU Bac+3 (1ère année)"; "DU Bac+4 (1ère année)"; "DUT (1ère
   ↪ année)";
8   "Diplôme étranger "; "L1 (1ère année de licence)";
9   "LP1 (1ère année de licence professionnelle)"; "Non"; "PACES
   ↪ (1ère année)";
10  "PASS (1ère année)"; "PREPA Bac+1 (1ère année)"]

```

223 (VÉRIFICATIONS 215) + ≡ OCaml

```

1 and verification_description_numero_colonne liste_possibles
   ↪ formulaire =
2   let decoupage_description_annee x =
3     Re.matches
4       (Re.compile
5         (Re.alt
6           [Re.shortest (Re.seq [Re.bol;
7             Re.str "Diplôme étranger |";
8             Re.rep Re.any;
9             Re.str "- Bac+";
10            Re.digit;
11            Re.eol]);
12          Re.shortest (Re.seq [Re.bol;
13            Re.str "Autre diplôme français |";
14            Re.rep Re.any;
15            Re.str "- Autre diplôme Bac+";
16            Re.digit;
17            Re.opt (Re.seq [Re.str " à Bac+";
18              Re.digit]);
19            Re.eol]);
20          Re.shortest (Re.seq [Re.bol;
21            Re.str "Aucun cursus post-bac ni
   ↪ diplômé déclaré |";
22            Re.rep Re.any;
23            Re.eol]);
24          Re.shortest (Re.seq [Re.bol;
25            Re.rep (Re.compl [Re.set "|"]);
26            Re.eol]);
27          (* Re.shortest (Re.seq [Re.bol;Re.rep Re.any;Re.eol])
   ↪ *)
28        ])) x

```

```

29 in
30 List.fold_left
31   ( $\lambda$ acc  $\rightarrow$ 
32      $\lambda$ x  $\rightarrow$ 
33     acc && (List.mem
34       (List.hd
35         (String.split_on_char '|'
36           (List.hd
37             (decoupage_description_annee
38               (List.nth x numero_colonne))))))
39     liste_possibles))
40   true formulaire
41 in
42 assert((verification_description 42
43   description_annee_l3_autorisee formulaire)
44   && (verification_description 57
45     description_annee_l2_autorisee formulaire)
46   && (verification_description 72
47     description_annee_l1_autorisee formulaire));

```

224 Année Bac+4

- 88. Année bac+4
- 89. Année universitaire / civile (si calendrier austral)
- 90. Mention / spécialité / description du diplôme / formation dans le cas de diplôme étranger / Autre
- 91. Parcours
- 92. Césure ?
- 93. Pas de moyenne au semestre ?
- 94. Moyenne au premier semestre
- 95. Moyenne au second semestre
- 96. Relevés de notes au premier semestre
- 97. Relevés de notes au second semestre
- 98. Relevé(s) de notes traduits ?
- 99. Justification d'absence des relevés
- 100. Établissement
- 101. Aménagement d'études
- 102. Avez-vous suivi cette formation dans le cadre d'une mobilité internationale ?

225 Empiriquement, la case Année bac+4 contient

- 1 *"Aucun cursus post-bac ni diplômé déclaré "*; *"Autre diplôme*
 \hookrightarrow français ";
- 2 *"DU Bac+4 (4ème année)"; "Diplôme étranger "; "INGENIEUR (4ème*
 \hookrightarrow année)";
- 3 *"M1 (1ère année de master)"; "Non"; "PREPA Bac+4 (1ère année)"]*

```

1  ::inscription_annee_bac_plus_4
2  ::annee_annee_bac_plus_4
3  ::mention_annee_bac_plus_4
4  ::parcours_annee_bac_plus_4
5  ::_
6  ::_
7  ::moyenne_s1_annee_bac_plus_4
8  ::moyenne_s2_annee_bac_plus_4
9  ::_
10 ::_
11 ::_
12 ::_
13 ::etablissement_annee_bac_plus_4
14 ::_
15 ::_

```

226 Année Bac+5

103. Année bac+5
104. Année universitaire / civile (si calendrier austral)
105. Mention / spécialité / description du diplôme / formation dans le cas de diplôme étranger / Autre
106. Parcours
107. Césure ?
108. Pas de moyenne au semestre ?
109. Moyenne au premier semestre
110. Moyenne au second semestre
111. Relevés de notes au premier semestre
112. Relevés de notes au second semestre
113. Relevé(s) de notes traduits ?
114. Justification d'absence des relevés
115. Établissement
116. Aménagement d'études
117. Avez-vous suivi cette formation dans le cadre d'une mobilité internationale ?

(LIGNECANDIDAT 211) + ≡

OCaml

```

1  ::inscription_annee_bac_plus_5
2  ::annee_annee_bac_plus_5
3  ::mention_annee_bac_plus_5
4  ::parcours_annee_bac_plus_5
5  ::_
6  ::_
7  ::moyenne_s1_annee_bac_plus_5
8  ::moyenne_s2_annee_bac_plus_5
9  ::_
10 ::_
11 ::_
12 ::_

```

13 ::etablissement_annee_bac_plus_5
 14 ::_
 15 ::_

227 Empiriquement, la case Année bac+5 contient

1 ["Aucun cursus post-bac ni diplômé déclaré "; "DECG Bac+5 (5ème
 ↪ année)";
 2 "Diplôme étranger "; "INGENIEUR (5ème année)"; "M2 (2ème année de
 ↪ master)";
 3 "Non"; "PREPA Bac+5 (1ère année)"]

228 Au-delà du Bac+5

- 118. Année au-delà du bac+5
- 119. Année universitaire / civile (si calendrier austral)
- 120. Mention / spécialité / description du diplôme / formation dans le cas de diplôme étranger / Autre
- 121. Parcours
- 122. Césure ?
- 123. Pas de moyenne au semestre ?
- 124. Moyenne au premier semestre
- 125. Moyenne au second semestre
- 126. Relevés de notes au premier semestre
- 127. Relevés de notes au second semestre
- 128. Relevé(s) de notes traduits ?
- 129. Justification d'absence des relevés
- 130. Établissement
- 131. Aménagement d'études
- 132. Avez-vous suivi cette formation dans le cadre d'une mobilité internationale ?

{LIGNECANDIDAT 211} + ≡

OCaml

1 ::_
 2 ::_
 3 ::_
 4 ::_
 5 ::_
 6 ::_
 7 ::_
 8 ::_
 9 ::_
 10 ::_
 11 ::_
 12 ::_
 13 ::_
 14 ::_
 15 ::_

229 Empiriquement, les cases des items 43, 58, 73, 88, 103 and 118 contenant le type des années (bac+3, bac+2, bac+1, bac+4, bac+5 et au-delà de bac+5) contiennent soit une des chaînes de caractères :

- BACHELOR (1ère année)
- BACHELOR (2ème année)
- BACHELOR (3ème année)
- Bachelor+4 (1ère année)
- Bachelor+4 (2ème année)
- Bachelor+4 (3ème année)
- BTS (1ère année)
- BTS (2ème année)
- BUT1 (1ère année de bachelor universitaire de technologie)
- BUT2 (2ème année de bachelor universitaire de technologie)
- BUT3 (3ème année de bachelor universitaire de technologie)
- CPGE Bac+2 (1ère année)
- CPGE Bac+2 (2ème année)
- DCG (1ère année)
- DCG (2ème année)
- DCG (3ème année)
- DECG Bac+5 (5ème année)
- DEUG (2ème année)
- Diplôme d'Etat Bac+2 (2ème année)
- Diplôme d'Etat Bac+3 (3ème année)
- DOCT MED (1ère année)
- DU Bac+1 (1ère année)
- DU Bac+2 (1ère année)
- DU Bac+2 (2ème année)
- DU Bac+3 (1ère année)
- DU Bac+3 (2ème année)
- DU Bac+3 (3ème année)
- DU Bac+4 (1ère année)
- DU Bac+4 (4ème année)
- DU Bac+5 (2ème année)
- DUT (1ère année)
- DUT (2ème année)
- INGENIEUR (3ème année)
- INGENIEUR (4ème année)
- INGENIEUR (5ème année)
- L1 (1ère année de licence)
- L2 (2ème année de licence)
- L3 (3ème année de licence)

- M1 (1ère année de master)
- M2 (2ème année de master)
- MS (1ère année)
- MS (2ème année)
- LP1 (1ère année de licence professionnelle)
- LP2 (2ème année de licence professionnelle)
- LP3 (3ème année de licence professionnelle)
- Non
- PACES (1ère année)
- PASS (1ère année)
- PREPA Bac+1 (1ère année)
- PREPA Bac+4 (1ère année)
- PREPA Bac+5 (1ère année)

ou bien

- Aucun cursus post-bac ni diplômé déclaré suivi de " | " et un texte libre entré par le candidat·e (pouvant contenir des retours à la ligne) ;
- Autre diplôme français suivi de " | ", d'un texte libre (pouvant contenir des retours à la ligne) entré par le candidat·e et de "- Autre diplôme Bac+", d'un chiffre, de " à Bac+" et un autre chiffre.
- Diplôme étranger suivi de " | ", d'un texte libre (pouvant contenir des retours à la ligne) entré par le candidat·e et de "- Bac+" et un chiffre.

Si la case contient Non ou une ligne commençant par Aucun cursus post-bac ni diplômé déclaré, les autres colonnes de la section sont vides.

- ¹ ["Aucun cursus post-bac ni diplômé déclaré "; "Diplôme étranger ";
² "MS (1ère année)"; "MS (2ème année)"; "Non"]

230 Récapitulatif

133. Suivez-vous une formation actuellement ?
134. Quel est votre dernier Etablissement d'inscription ?
135. Quelle formation avez-vous suivie au 1er semestre de vos études supérieures ?
136. Quelle moyenne avez-vous obtenue à ce semestre 1 ?
137. Quelle formation avez-vous suivie au 2ème semestre de vos études supérieures ?
138. Quelle moyenne avez-vous obtenue à ce semestre 2 ?
139. Quelle formation avez-vous suivie au 3ème semestre de vos études supérieures ?
140. Quelle moyenne avez-vous obtenue au semestre 3 ?
141. Quelle formation avez-vous suivie au 4ème semestre de vos études supérieures ?
142. Quelle moyenne avez-vous obtenue au semestre 4 ?
143. Quelle formation avez-vous suivie au 5ème semestre de vos études supérieures ?
144. Quelle moyenne avez-vous obtenue au semestre 5 ?

145. Quelle formation avez-vous suivie au 6ème semestre de vos études supérieures?
146. Quelle moyenne avez-vous obtenu au semestre 6?
147. Possédez-vous une Certification de langue étrangère? Si oui, laquelle?
148. Indiquer l'éventuel score obtenu au certificat de langue
149. Avez-vous réalisé d'une mobilité internationale dans le cadre de vos études (hors stage)?
150. Si vous avez eu des expériences professionnelles dans le cadre de vos études (stage et/ou alternance), indiquez leur durée cumulée en mois

(LIGNECANDIDAT 211) + ≡

OCamI

1 ::_
 2 ::_
 3 ::_
 4 ::_
 5 ::_
 6 ::_
 7 ::_
 8 ::_
 9 ::_
 10 ::_
 11 ::_
 12 ::_
 13 ::_
 14 ::_
 15 ::_
 16 ::_
 17 ::_
 18 ::_

231 (CONSTRUCTIONCANDIDAT) ≡

OCamI

```

1 let (notesp1,_,_) =
2   meilleure_annee_de_licence
3   (annees_universitaires inscription_annee_bac_plus_1
4     ↪ annee_annee_bac_plus_1
5     mention_annee_bac_plus_1 parcours_annee_bac_plus_1
6     ↪ moyenne_s1_annee_bac_plus_1
7     moyenne_s2_annee_bac_plus_1 etablissement_annee_bac_plus_1)
8 and (notesp2,_,_) =
9   meilleure_annee_de_licence
10  (annees_universitaires inscription_annee_bac_plus_2
11    ↪ annee_annee_bac_plus_2
12    mention_annee_bac_plus_2 parcours_annee_bac_plus_2
13    ↪ moyenne_s1_annee_bac_plus_2
14    moyenne_s2_annee_bac_plus_2 etablissement_annee_bac_plus_2)
15 and (notesp3,autre_l3,upec_l3) =
16   meilleure_annee_de_licence
17   (annees_universitaires inscription_annee_bac_plus_3
18     ↪ annee_annee_bac_plus_3

```

```

14     mention_annee_bac_plus_3 parcours_annee_bac_plus_3
      ↪ moyenne_s1_annee_bac_plus_3
15     moyenne_s2_annee_bac_plus_3 etablisement_annee_bac_plus_3)
16 and (m1,_) =
17     annee_master
18     (annees_universitaires inscription_annee_bac_plus_4
      ↪ annee_annee_bac_plus_4
19     mention_annee_bac_plus_4 parcours_annee_bac_plus_4
      ↪ moyenne_s1_annee_bac_plus_4
20     moyenne_s2_annee_bac_plus_4 etablisement_annee_bac_plus_4)
21 and (m2,_) =
22     annee_master
23     (annees_universitaires inscription_annee_bac_plus_5
      ↪ annee_annee_bac_plus_5
24     mention_annee_bac_plus_5 parcours_annee_bac_plus_5
      ↪ moyenne_s1_annee_bac_plus_5
25     moyenne_s2_annee_bac_plus_5 etablisement_annee_bac_plus_5)
26 in
27 (numero,{nom = nom;
28     prenom = prenom;
29     adresse_email = adresse_email;
30     baccaureat = baccaureat_of_type_caracteristiques
31                 type_baccaureat serie_baccaureat;
32     bac_plus_1 = notesp1;
33     bac_plus_2 = notesp2;
34     bac_plus_3 = notesp3;
35     autre_licence = autre_l3;
36     upec = upec_l3;
37     bac_plus_4 = m1;
38     bac_plus_5 = m2;
39 })

```

c.2 Construction d'une candidate

232 Les structures de données représentant les disciplines, notes et établissement sont des types sommes, le type représentant une année universitaire est plus complexe : selon les types d'années, on récolte plus ou moins d'information.

233 Chaque année universitaire a une année, qui peut être deux années civiles séparées par un trait d'union (dans le cas d'une année universitaire boréale) ou une année civile unique. Dans le premier cas, on prend l'année de fin.

⟨CONSTRUCTIONLICENCE⟩ ≡ OCaml

```

1 let annee_of_string (annee : string) : int =
2   try List.fold_left max 0
3     (List.map int_of_string
4      (String.split_on_char '-' annee))
5   with int_of_string → failwith annee

```

234 On ne considère qu'une discipline : le droit. Toutes les autres sont regroupées.

⟨CARACTÉRISTIQUESANNÉEUNIVERSITAIRE⟩ ≡ OCaml

```
1 type discipline = Droit of string | Autre of string
```

- 235 On reconnaît qu'une année est une année de droit si l'intitulé de la mention commence par « Droit » (cas d'une licence générale) ou « Carrières juridiques » (cas d'un BUT).

⟨CONSTRUCTIONLICENCE 233⟩ +≡ OCaml

```
1 let discipline_of_string (d : string) : discipline =
2   if (String.starts_with ~prefix:"Droit" d)
3     || (String.starts_with ~prefix:"droit" d)
4     || (String.starts_with ~prefix:"Carrières juridiques" d)
5   then Droit(d)
6   else Autre(d)
```

- 236 Pour chaque semestre d'une année, il peut y avoir une note.

⟨CARACTÉRISTIQUESANNÉEUNIVERSITAIRE 234⟩ +≡ OCaml

```
1 type note = float option
```

- 237 MonMaster utilise parfois 99 pour indiquer qu'un champ n'est pas rempli : on coupe donc les notes à 20, et considère qu'une note au-dessus de 20, ou n'étant pas un nombre, est absente.

⟨CONSTRUCTIONLICENCE 233⟩ +≡ OCaml

```
1 let note_of_string (s : string) : note =
2   try
3     let note = float_of_string s
4     in if note > 0. && note <= 20.
5       then Some note
6       else None
7   with float_of_string → None
```

- 238 On considère trois types d'établissement : l'université Paris est Créteil, les instituts catholiques, et tous les autres.

⟨CARACTÉRISTIQUESANNÉEUNIVERSITAIRE 234⟩ +≡ OCaml

```
1 type etablissement = UPEC | Catholique of string
2                   | AutreEtablissement of string
3 exception PasEtablissement
4 exception Catholique
```

- 239 ⟨CONSTRUCTIONLICENCE 233⟩ +≡ OCaml

```
1 let contains (s1 : string) (s2 : string) : bool =
2   let re = Re.compile (Re.str s2)
3   in
4   try let _ = (Re.exec re s1) in true
5   with Not_found → false
```

- 240 ⟨CONSTRUCTIONLICENCE 233⟩ +≡ OCaml

```

1 let etablisement_of_string (etab : string) : etablisement =
2   if (String.starts_with ~prefix:"Université Paris-XII (Paris 12)
   ↪ (Université Paris Est Créteil Val-de-Marne - UPEC) - 94010 -
   ↪ Créteil" etab)
3   then UPEC
4   else if String.equal etab ""
5   then raise PasEtablissement
6   else if (contains etab "catholique") || (contains etab
   ↪ "Catholique")
7   then raise Catholique
8   else AutreEtablissement(etab)

```

241 <STRUCTURES> ≡

OCaml

```

1 open CalculScoresCasNominal
2   <CARACTÉRISTIQUESANNÉEUNIVERSITAIRE 234>
3 type annee_universitaire =
4   L1 of int × discipline × note × note × etablisement
5   | L2 of int × discipline × note × note × etablisement
6   | L3 of int × discipline × note × note × etablisement
7   | LP1 of int × discipline
8   | LP2 of int × discipline
9   | LP3 of int × discipline
10  | BUT1 of int × discipline × note × note
11  | BUT2 of int × discipline × note × note
12  | BUT3 of int × discipline × note × note
13  | M1 of int × discipline
14  | M2 of int × discipline
15  | Etranger of string × int × discipline
16  | Autre of string × int × discipline
17  | Aucun

```

242 La première étape de construction consiste à évaluer le baccalauréat du candidat. Comme constaté au § 216, toutes les spécialités du baccalauréat ne sont pas présentes dans le fichier source, aussi, nous ne traitons que celles que nous avons constaté.

<CONSTRUCTIONBACCALAURÉAT> ≡

OCaml

```

1 let specialite_of_string (s : string) : SpecialiteGeneral.t =
2   match s with
3     | "Théâtre (ARTS_THEATRE)"
4     | "Danse (ARTS_DANSE)"
5     | "Arts plastiques (ARTS_ARTS PLA)"
6       → Arts
7     | "Histoire géographie, géopolitique et sciences politiques
   ↪ (HGGSP)"
8       → HistoireGeographieGeopolitiqueEtSciencesPolitiques
9     | "Humanités, littérature et philosophie (HLP)"
10      → HumanitesLitteraturePhilosophie
11     | "Mathématiques (MATHS)"
12      → Mathematiques
13     | "Numérique et sciences informatiques (NSI)"

```

```

14     → NumeriqueEtSciencesInformatiques
15 | "Physique-chimie (PH_CH)"
16     → PhysiqueChimie
17 | "Sciences de la vie et de la Terre (SVT)"
18     → SciencesDeLaVieEtDeLaTerre
19 | "sciences économiques et sociales (SES)"
20     → SciencesEconomiquesEtSociales
21 | "LLCER AMC – Langues, littératures et cultures étrangères et
    ↪ régionales Anglais monde contemporain"
22 | "LLCER Américain – Langues, littératures et cultures étrangères
    ↪ et régionales Américain"
23 | "LLCER Anglais – Langues, littératures et cultures étrangères et
    ↪ régionales Anglais"
24 | "LLCER Espagnol – Langues, littératures et cultures étrangères
    ↪ et régionales Espagnol"
25     → LanguesLitteraturesEtCulturesEtrangeres
26 | _ → failwith s

```

243 La fonction sera à attendre si une des spécialités

- **BiologieEcologie**,
- **EducationPhysiquePratiquesCultureSportives**,
- **LitteraturesLanguesEtCulturesAntiquite**
- **SciencesDeLingenieur**

apparaissent, ainsi que d'autres variantes de LLCER ou d'Arts.

244 On se donne une représentation canonique des spécialités influant la note.

(ÉCRITUREBACCALAURÉAT) ≡ OCaml

```

1 let string_of_specialite (s : SpecialiteGeneral.t) : string =
2   match s with
3     NumeriqueEtSciencesInformatiques → "Numérique et sciences
    ↪ informatiques"
4   | Mathematiques → "Mathématiques"
5   | SciencesDeLingenieur → "Sciences de l'ingénieur"
6   | PhysiqueChimie → "Physique-Chimie"
7   | SciencesDeLaVieEtDeLaTerre → "Sciences de la vie et de la
    ↪ Terre"
8   | BiologieEcologie → "Biologie Écologie"
9   | SciencesEconomiquesEtSociales → "Sciences économiques et
    ↪ sociales"
10  | _ → ""

```

245 (ÉCRITUREBACCALAURÉAT 244) + ≡ OCaml

```

1 let string_of_baccalaureat (b : CaracteristiquesBac.t) : string =
2   match b with
3     GeneralSeries(Scientifique) → "\\textsc{s}"
4   | GeneralSeries(EconomiqueEtSocial) → "\\textsc{es}"
5   | GeneralSpecialites(s1,s2) →
6     let ss = (string_of_specialite s1) ^ " " ^
    ↪ (string_of_specialite s2) in

```

```

7     if (String.length ss > 1)
8     then
9         "spécialité " ^ ss
10    else ""
11    | EquivalenceEvaluatede(_) → "Équivalence"
12    | Capacite → "Capacité"
13    | _ → ""

```

246 <CONSTRUCTIONBACCALAURÉAT 242> + ≡

OCaml

```

1  let baccalaureat_of_type_caracteristiques
2      (b : string) (caracteristiques : string)
3      : CaracteristiquesBac.t =
4  match b with
5      "Général" →
6          (match caracteristiques with
7              "Scientifique (S)"
8              | "E" → GeneralSeries(Scientifique)
9              | "Economique et social (ES)" →
10                 GeneralSeries(EconomiqueEtSocial)
11              | "Littéraire (L)" →
12                 GeneralSeries(Litteraire)
13              | _ → (match (String.split_on_char '|' caracteristiques)
14                  with s1::s2::[] →
15                      GeneralSpecialites(specialite_of_string s1,
16                                          specialite_of_string s2)
17                  | _ → failwith (b ^ caracteristiques)))
18  | "Technologique" →
19      (match caracteristiques with
20          "Sciences et technologies du management et de la gestion
21          ↪ (STMG)"
22          → Technologique(STMG)
23          | "Sciences et technologies de la santé et du social (ST2S)"
24          → Technologique(ST2S)
25          | "Sciences et technologies de l'industrie et du développement
26          ↪ durable (STI2D)"
27          → Technologique(STI2D)
28          | "Sciences et technologies de laboratoire (STL)"
29          → Technologique(STL)
30          | _ → failwith caracteristiques)
31  | "Professionnel" →
32      (match caracteristiques with
33          "Bacs professionnels tertiaires" →
34              Professionnel
35          | _ → failwith caracteristiques)
36  | "CAPACITE DE DROIT" → Capacite
37  | "TITRE FRANCAIS ADMIS EN DISPENSE"
38  | "TITRE ETRANGER ADMIS EN EQUIVALENCE (dont bac
39  ↪ franco-allemand, Abibac, Bachibac et Esabac) "
40  | "TITRE ETRANGER ADMIS EN EQUIVALENCE (dont bac
41  ↪ franco-allemand, Abibac, Bachibac et Esabac)"

```

```

38 | "VALIDATION D'ETUDES, D'EXPERIENCES PROFESSIONNELLES, D'ACQUIS
    ↪ PERSONNELS"
39 | "AUTRE CAS DE NON BACHELIERS"
40 | "PROMOTION SOCIALE" | "" → Equivalence
41 | _ → failwith b

```

247 <CONSTRUCTIONLICENCE 233> +≡

OCaml

```

1 let rec zip7 l1 l2 l3 l4 l5 l6 l7 =
2   match l1,l2,l3,l4,l5,l6,l7 with
3   | t1::q1,t2::q2,t3::q3,Some (t4::q4),t5::q5,t6::q6,
4     t7::q7 →
5     (t1,t2,t3^" ^t4,t5,t6,t7)::
6       zip7 q1 q2 q3 (Some q4) q5 q6 q7
7   | t1::q1,t2::q2,t3::q3,None,t5::q5,t6::q6,
8     t7::q7 →
9     (t1,t2,t3,t5,t6,t7)::
10      zip7 q1 q2 q3 None q5 q6 q7
11 | [],[],[],Some [],[],[],[] → []
12 | _ → failwith "pas la même longueur"

```

248 <CONSTRUCTIONLICENCE 233> +≡

OCaml

```

1 let annees_universitaires
2   (inscription : string) (annee : string) (mention : string)
3   (parcours : string) (moyenne_s1 : string) (moyenne_s2 :
4     ↪ string)
5   (etablissement : string) : annee_universitaire list =
6   if (String.starts_with ~prefix:"Aucun cursus post-bac ni diplômé
7     ↪ déclaré" inscription)
8     || (String.equal "Non" inscription)
9   then [Aucun]
10  else try
11    (List.map
12     (λ(x,y,z,s1,s2,etablissement) →
13       try
14         (match x with
15         | "L1 (1ère année de licence)" →
16           L1(annee_of_string y,disipline_of_string z,
17             note_of_string s1, note_of_string s2,
18             etablissement_of_string etablissement)
19         | "L2 (2ème année de licence)" →
20           L2(annee_of_string y,disipline_of_string z,
21             note_of_string s1, note_of_string s2,
22             etablissement_of_string etablissement)
23         | "L3 (3ème année de licence)" →
24           L3(annee_of_string y,disipline_of_string z,
25             note_of_string s1, note_of_string s2,
26             etablissement_of_string etablissement)
27         | "M1 (1ère année de master)" →
28           (match note_of_string s1,note_of_string s2 with
29             None,None → Aucun

```

```

28         | _ → M1(annee_of_string y, discipline_of_string
           ↪ z)
29 | "M2 (2ème année de master)" →
30   (match note_of_string s1, note_of_string s2 with
31     None, None → Aucun
32     | _ → M2(annee_of_string y, discipline_of_string
           ↪ z))
33 | "LP1 (1ère année de licence professionnelle)" →
34   LP3(annee_of_string y, discipline_of_string z)
35 | "LP2 (2ème année de licence professionnelle)" →
36   LP3(annee_of_string y, discipline_of_string z)
37 | "LP3 (3ème année de licence professionnelle)" →
38   LP3(annee_of_string y, discipline_of_string z)
39 | "BUT1 (1ère année de bachelor universitaire de
     ↪ technologie)" →
40   BUT1(annee_of_string y, discipline_of_string z,
41         note_of_string s1, note_of_string s2)
42 | "BUT2 (2ème année de bachelor universitaire de
     ↪ technologie)" →
43   BUT2(annee_of_string y, discipline_of_string z,
44         note_of_string s1, note_of_string s2)
45 | "BUT3 (3ème année de bachelor universitaire de
     ↪ technologie)" →
46   BUT3(annee_of_string y, discipline_of_string z,
47         note_of_string s1, note_of_string s2)
48 | _ → if (String.starts_with ~prefix:"Diplôme
     ↪ étranger" inscription)
49     then Etranger(inscription, annee_of_string
     ↪ y, discipline_of_string z)
50     else Autre(inscription, annee_of_string
     ↪ y, discipline_of_string z))
51 with PasEtablissement → Aucun
52 | Catholique → Etranger(inscription, annee_of_string
     ↪ y, discipline_of_string z)
53 )
54 (zip7
55   (List.fold_right
56     (λy →
57       λx →
58         if not
59           ⟨DÉBUTDIPLÔMES 249⟩
60         then x
61         else y::x)
62     (String.split_on_char '\n' inscription)
63     [])
64   )
65   (String.split_on_char '\n' annee)
66   (String.split_on_char '\n' mention)
67   (match (String.split_on_char '\n' parcours) with
68     [] → None
69     | p → Some p)

```

```

70         (String.split_on_char '\n' moyenne_s1)
71         (String.split_on_char '\n' moyenne_s2)
72         (String.split_on_char '\n' etablissement)))
73     with _ → print_string(inscription ^ annee ^ mention ^
74                 parcours ^ moyenne_s1 ^ moyenne_s2 ^
75                 etablissement);
76     print_newline ();
77     print_newline();
78     []

```

249 <DÉBUTDIPLÔMES> ≡

OCaml

```

1  (List.fold_left
2    (λb →
3      λprefixe →
4        b || (String.starts_with ~prefix:prefixe y))
5    false
6    ["L"; "M1"; "M2"; "LP"; "BUT"; "BTS"; "DU"; "PASS";
7     "BACHELOR"; "Bachelor"; "DOCT MED"; "PACES"; "CPGE";
8     "PREPA"; "DEUG"; "DCG"; "DECG"; "INGENIEUR"; "Diplôme d'Etat";
9     "Diplôme étranger"; "Autre diplôme français"])

```

250 <CONSTRUCTIONLICENCE 233> +≡

OCaml

```

1  let meilleure_annee_de_licence l =
2    List.fold_left
3      (λ(l',autre,upec) →
4        λl →
5          match l,l' with
6            L1(a,Droit(x),Some(n1),Some(n2),etab),
7            L1(a',Droit(x'),Some(n1'),Some(n2'),etab') →
8              if a > a'
9                then (L1(a,Droit(x),Some(n1),Some(n2),etab),autre,upec)
10             else
11                 ⇨ (L1(a',Droit(x'),Some(n1'),Some(n2'),etab'),autre,upec)
12             | _,L1(a,Droit(x),_,_,_) → (l',autre,upec)
13             | L1(a,Droit(x),_,_,_),_ → (l,autre,upec)
14             | L2(a,Droit(x),Some(n1),Some(n2),etab),
15             L2(a',Droit(x'),Some(n1'),Some(n2'),etab') →
16               if a > a'
17                 then (L2(a,Droit(x),Some(n1),Some(n2),etab),autre,upec)
18             else
19                 ⇨ (L2(a',Droit(x'),Some(n1'),Some(n2'),etab'),autre,upec)
20             | _,L2(a,Droit(x),_,_,_) → (l',autre,upec)
21             | L2(a,Droit(x),_,_,_),_ → (l,autre,upec)
22             | L3(a,Droit(x),Some(n1),n2,etab),
23             L3(a',Droit(x'),Some(n1'),n2',etab') →
24               if a > a'
25                 then (L3(a,Droit(x),Some(n1),n2,etab),autre,etab = UPEC)
26             else (L3(a',Droit(x'),Some(n1'),n2',etab'),autre,etab' =
27                 ⇨ UPEC)
28             | _,L3(a,Droit(x),_,_,etab) → (l',autre,etab = UPEC)

```

```

26 | L3(a,Droit(x),_,_,etab),_ → (l,autre,etab=UPEC)
27 | _,L3(a,Autre(_),_,_,_) → (l',true,upec)
28 | L3(a,Autre(_),_,_,_),_ → (l,true,upec)
29 | Etranger(_,_ ,Droit(x)),_ → l,autre,upec
30 | _,Etranger(_,_ ,Droit(x)) → l',autre,upec
31 | BUT1(a,Droit(x),_,_,_) → l,autre,upec
32 | BUT2(a,Droit(x),_,_,_) → l,autre,upec
33 | BUT3(a,Droit(x),_,_,_) → l,autre,upec
34 | _,_ → l',autre,upec)
35 (Aucun, false, false) l

```

251 $\langle \text{CONSTRUCTIONLICENCE } 233 \rangle + \equiv$

OCaml

```

1 let annee_master l =
2   List.fold_left
3     (λ(acc1,acc2) ->
4       λm ->
5         match m with
6           M1(_,Droit(_)) → true, true
7           | M1(_,Autre(_)) → true, acc2
8           | M2(_,Droit(_)) → true, true
9           | M2(_,Autre(_)) → true, acc2
10          | _ → acc1,acc2
11        ) (false,false) l

```

D Manipulation des dossiers

D.1 Tri des dossiers en groupes

252 $\langle \text{FONCTIONS CALCUL } 45 \rangle + \equiv$

OCaml

```
1  $\langle \text{TRI DOSSIERS FORMULAIRES } 253 \rangle$ 
```

253 La fonction `tri_dossier` sépare les dossiers des candidat·es selon des caractéristiques objectives.

$\langle \text{TRI DOSSIERS FORMULAIRES} \rangle \equiv$

```

1 let tri_dossier (dossier : dossier_formulaire) : groupes_formulaire
2   = match dossier with
3      $\langle \text{CONDITION NOMINAL } 254 \rangle$ 
4     → Nominal
5   |
6      $\langle \text{CONDITION QUASI NOMINAL } 256 \rangle$ 
7     → QuasiNominal
8   |
9      $\langle \text{CONDITION NOMINAL SAISIE DÉFAILLANTE } 257 \rangle$ 
10    → NominalSaisieDefaillante
11  |
12     $\langle \text{CONDITION AUTRE LICENCE DROIT } 258 \rangle$ 
13    → AutreLicenceDroit
14  |

```

```

15     <CONDITIONBUTCARRIEREJURIDIQUES 259>
16     → BUTCarriereJuridiques
17     |
18     -
19     → Autre

```

- 254 La condition énoncée au § 28 pour être dans le cas nominal est :
- être titulaire d'un baccalauréat général ou technologique (ou d'une capacité en droit) français;
 - avoir préparé les cinq ou six premiers semestres d'une licence de droit préparée dans un établissement public à caractère scientifique, culturel et professionnel au sens de l'article L. 711-1 du Code de l'éducation.

```

<CONDITIONNOMINAL> ≡ OCaml
1   {
2     baccalaureat = GeneralSpecialites(_,_);
3     bac_plus_1 = L1(_,Droit(_),Some(_),Some(_),_);
4     bac_plus_2 = L2(_,Droit(_),Some(_),Some(_),_);
5     bac_plus_3 = L3(_,Droit(_),Some(_),_,_);
6   }
7   |
8   {
9     baccalaureat = GeneralSeries(_);
10    bac_plus_1 = L1(_,Droit(_),Some(_),Some(_),_);
11    bac_plus_2 = L2(_,Droit(_),Some(_),Some(_),_);
12    bac_plus_3 = L3(_,Droit(_),Some(_),_,_);
13  }
14  |
15  {
16    baccalaureat = Technologique(_);
17    bac_plus_1 = L1(_,Droit(_),Some(_),Some(_),_);
18    bac_plus_2 = L2(_,Droit(_),Some(_),Some(_),_);
19    bac_plus_3 = L3(_,Droit(_),Some(_),_,_);
20  }
21  |
22  {
23    baccalaureat = Capacite;
24    bac_plus_1 = L1(_,Droit(_),Some(_),Some(_),_);
25    bac_plus_2 = L2(_,Droit(_),Some(_),Some(_),_);
26    bac_plus_3 = L3(_,Droit(_),Some(_),_,_);
27  }

```

Étendu en §255.

- 255 De plus, les dossiers quasinominaux peuvent être traités comme des dossiers nominaux une fois qu'on a évalué leurs baccalauréats, cf § 134.

```

<CONDITIONNOMINAL 254> + ≡ OCaml
1   |
2   {
3     baccalaureat = EquivalenceEvaluee(_);
4     bac_plus_1 = L1(_,Droit(_),Some(_),Some(_),_);

```

```

5     bac_plus_2 = L2(_,Droit(_),Some(_),Some(_),_);
6     bac_plus_3 = L3(_,Droit(_),Some(_),_,_);
7 }

```

256 La condition énoncée au § 128 pour être dans le cas quasinominal est:

- être titulaire d’un baccalauréat étranger ou professionnel;
- avoir préparé les cinq ou six premiers semestres d’une licence de droit préparée dans un établissement public à caractère scientifique, culturel et professionnel au sens de l’article L. 711-1 du Code de l’éducation.

$\langle \text{CONDITIONQUASINOMINAL} \rangle \equiv$ OCaml

```

1 {
2     baccalaureat = Professionnel;
3     bac_plus_1 = L1(_,Droit(_),Some(_),Some(_),_);
4     bac_plus_2 = L2(_,Droit(_),Some(_),Some(_),_);
5     bac_plus_3 = L3(_,Droit(_),Some(_),_,_);
6 }
7 |
8 {
9     baccalaureat = Equivalence;
10    bac_plus_1 = L1(_,Droit(_),Some(_),Some(_),_);
11    bac_plus_2 = L2(_,Droit(_),Some(_),Some(_),_);
12    bac_plus_3 = L3(_,Droit(_),Some(_),_,_);
13 }

```

257 La condition énoncée au § 137 pour être dans le cas nominal avec saisie défailante est:

- avoir préparé les cinq ou six premiers semestres d’une licence de droit préparée dans un établissement public à caractère scientifique, culturel et professionnel au sens de l’article L. 711-1 du Code de l’éducation.

On ne prend en compte que le fait que le candidat-e ait déclaré dans le formulaire les trois années d’une licence de droit.

$\langle \text{CONDITIONNOMINALSAISIEDÉFAILLANTE} \rangle \equiv$ OCaml

```

1 {
2     bac_plus_1 = L1(_,Droit(_),_,_,_);
3     bac_plus_2 = L2(_,Droit(_),_,_,_);
4     bac_plus_3 = L3(_,Droit(_),_,_,_);
5 }

```

258 La condition énoncée au § 142 pour être dans le cas des autres licences de droit est appréciée par le fait que l’année bac+3 est une année de droit étrangère.

$\langle \text{CONDITIONAUTRELICENCEDROIT} \rangle \equiv$ OCaml

```

1 {
2     bac_plus_3 = Etranger(.,.,Droit(_));
3 }

```

259 La condition énoncée au § 145 pour être dans le cas nominal avec saisie défaillante est :

- avoir préparé les cinq ou six premiers semestres d'un bachelor universitaire de technologie spécialité Carrières juridiques.

```

(CONDITIONBUTCARRIEREJURIDIQUES) ≡ OCaml
1  {
2    (* bac_plus_1 = BUT1(_,Droit(_),_,_); *)
3    (* bac_plus_2 = BUT2(_,Droit(_),_,_); *)
4    bac_plus_3 = BUT3(_,Droit(_),_,_);
5  }

```

D.2 Mise en forme des dossiers (quasi)nominaux

```

260 (MISEENFORME) ≡ OCaml
1  let notes_licence_of_candidat c =
2    match c with
3    {
4      bac_plus_1 = L1(_,Droit(_),Some(s1),Some(s2),_);
5      bac_plus_2 = L2(_,Droit(_),Some(s3),Some(s4),_);
6      bac_plus_3 = L3(_,Droit(_),Some(s5),s6,_);
7    }
8    →
9    {ScoreSemestres_in.notes_in =
10     {note_semestre = (Q.of_float s1, Q.of_float s2,
11                      Q.of_float s3, Q.of_float s4,
12                      Q.of_float s5,
13                      match s6 with
14                        None → Absent
15                        | Some(s6) → Present (Q.of_float s6)}}
16   }
17   | _ → failwith "Candidat nominal attendu";

```

```

261 (MISEENFORME 260) + ≡ OCaml
1  let construit_candidat_final (c : dossier_formulaire)
2    (note_lettre : int) (note_dossier : int) :
3    ⇨ CandidatCompletNominal.t =
4    match c with
5    {
6      bac_plus_1 = L1(_,Droit(_),Some(s1),Some(s2),_);
7      bac_plus_2 = L2(_,Droit(_),Some(s3),Some(s4),_);
8      bac_plus_3 = L3(_,Droit(_),Some(s5),s6,_);
9      autre_licence;
10     upec;
11     bac_plus_4;
12     bac_plus_5;
13     baccalaureat;
14   }
15   →
16   {

```

```

16     notes =
17         {note_semestre = (Q.of_float s1, Q.of_float s2,
18                         Q.of_float s3, Q.of_float s4,
19                         Q.of_float s5,
20                         match s6 with
21                             None → Absent |
22                             Some(s6) → Present (Q.of_float s6));
23     baccalaureat = baccalaureat;
24     master_annees_validees =
25         Z.of_int (if bac_plus_5 then 2
26                 else if bac_plus_4 then 1
27                 else 0);
28     licence_obtenue = autre_licence;
29     candidat_e_UPEC = upec;
30     appreciation_lettre = Z.of_int note_lettre;
31     appreciation_dossier = Z.of_int note_dossier
32 }
33 | _ → failwith "Candidat nominal attendu";

```

D.3 Autres fonctions de lecture

```

262 ⟨LIGNE|NVAIDE⟩ ≡ OCaml
1  (let rec vue x acc = match x with
2      [] → acc
3      | t::q → vue q (acc ^ " | " ^ t)
4  in failwith ("La ligne\n" ^ (vue ligne "") ^ "\n n'a pas assez de
   ↪ colonnes."))

```

E Fonctions d'écriture de fichiers de travail

E.1 Fiches de vérification

```

263 ⟨FICHEVÉRIFICATION⟩ ≡ OCaml
1  ⟨ÉCRITUREBACCALAURÉAT 244⟩
2  let fiche_verification c =
3      match c with
4          numero,
5          {nom;
6          prenom;
7          bac_plus_1 = L1(_,Droit(_),Some(s1),Some(s2),_);
8          bac_plus_2 = L2(_,Droit(_),Some(s3),Some(s4),_);
9          bac_plus_3 = L3(_,Droit(_),Some(s5),s6,_);
10         autre_licence;
11         baccalaureat;
12         upec;
13         bac_plus_4;
14         bac_plus_5;}
15     →
16     let oc = Stdlib.open_out

```

```

17      ( <REPertoireTRAVAILSORTIES 188> ^
      ↪ "mail_verification_candidats/"
18      ^ numero ^ ".tex") in
19  Printf.fprintf oc <PRÉAMBULEFICHEVÉRIFICATION 264>
20  Printf.fprintf oc <LICENCEFICHEVÉRIFICATIONUN 265>
21  (match s6 with
22    Some(s6') → Printf.fprintf oc
23      "&||textcolor{gray}{||textsc{s6:}}
      ↪ ||num{%.2f}" s6'
24    | _ → ());
25  Printf.fprintf oc <LICENCEFICHEVÉRIFICATIONDEUX 266>
26  Printf.fprintf oc <BACCALAURÉATFICHEVÉRIFICATION 267>
27  Printf.fprintf oc <AUTREDIPLÔMESFICHEVÉRIFICATION 268>
28  Printf.fprintf oc "||noindent ||textcolor{gray}{||Large Licence
  ↪ préparée à l' ||textsc{upec}}\n\n";
29  Printf.fprintf oc "||indent %s\n\n" (if upec then "oui" else
  ↪ "non");
30  Printf.fprintf oc "{||tiny ||noindent Cette propriété est vraie
  ↪ d'une ||textsc{l}3 de droit préparée à
  ↪ l' ||textsc{upec}.}\n||";
31  Printf.fprintf oc "||bigskip\n\n";
32  Printf.fprintf oc "||end{document}\n\n";
33  StdLib.close_out oc
34  | _ → ());

```

```

264 <PRÉAMBULEFICHEVÉRIFICATION> ≡ OCaml
  1 "||documentclass{lucart}\n\n";
  2 "||begin{document}\n\n";
  3 "{||noindent||LARGE ||MakeTitlecase{%s}
  ↪ ||scshape||MakeTitlecase{||MakeLowercase{%s}}||||" prenom nom;
  4 "||bigskip\n\n";

```

```

265 <LICENCEFICHEVÉRIFICATIONUN> ≡ OCaml
  1 "||noindent ||textcolor{gray}{||Large Licence de droit dans une
  ↪ université publique française}\n\n";
  2 "||||";
  3 "||begin{tabular}{lll}\n\n";
  4 "||textcolor{gray}{||textsc{l}1}&||textcolor{gray}{||textsc{s1:}}
  ↪ ||num{%.2f}&||textcolor{gray}{||textsc{s2:}} ||num{%.2f}||||\n\n"
  ↪ s1 s2;
  5 "||textcolor{gray}{||textsc{l}2}&||textcolor{gray}{||textsc{s3:}}
  ↪ ||num{%.2f}&||textcolor{gray}{||textsc{s4:}} ||num{%.2f}||||\n\n"
  ↪ s3 s4;
  6 "||textcolor{gray}{||textsc{l}3}&||textcolor{gray}{||textsc{s5:}}
  ↪ ||num{%.2f}" s5 ;

```

```

266 <LICENCEFICHEVÉRIFICATIONDEUX> ≡ OCaml
  1 "\n||end{tabular}\n\n";
  2 "||vline\n\n";

```

```

3  "||||n||||n";
4  "{|tiny En cas de redoublement, on ne considère que les notes de
   ↪ l'année validée.}|n||||";
5  "\\bigskip\n\n";

```

267 <BACCALAURÉATFICHEVÉRIFICATION> ≡ OCaml

```

1  "\\noindent \\textcolor{gray}{\\Large Baccalauréat}\\n\n";
2  "\\indent %s |||" (string_of_baccalaureat baccalaureat);
3  "{|tiny \\noindent Les notes du baccalauréat ne sont pas prises en
   ↪ compte.}|n||||";
4  "\\bigskip\n\n";

```

268 <AUTREDIPLÔMESFICHEVÉRIFICATION> ≡ OCaml

```

1  "\\noindent \\textcolor{gray}{\\Large Autres diplômes nationaux dans
   ↪ une université publique française}\\n\n";
2  "\\indent Licence: %s\\n\n" (if autre_licence then "licence" else
   ↪ "non");
3  "\\indent \\textsc{m}1: %s\\n\n" (if bac_plus_4 then "\\textsc{m1}"
   ↪ else "non");
4  "\\indent \\textsc{m}2: %s\\n\n" (if bac_plus_5 then "\\textsc{m2}"
   ↪ else "non");
5  "\\bigskip\n\n";

```

E.2 Liasses

269 <FONCTIONS CALCUL 45> + ≡

```

1  <HISTOGRAMMES 270>

```

270 <HISTOGRAMMES> ≡ OCaml

```

1  let remplissage y h pas fin =
2    for i = 0 to fin
3    do
4      (if y ≥ (float_of_int (pas × i))
5       && y < (float_of_int (pas × (i+1))))
6      then h.(i) ← h.(i) + 1
7      else ()
8    done
9  let histogrammes candidats_score =
10   List.fold_left
11     (λ(h_semestres, h_baccalaureat, h_autres_diplomes,
12      h_upec, h_appreciation, h_total)
13      → λ(_, (_, score, _))
14      → let (y_semestres, y_baccalaureat, y_autres_diplomes,
15            y_upec, y_appreciation, y_total)
16          = (match score with
17             {ScoreCandidat.points_licence;
18              points_bac;
19              points_diplomes;
20              points_dossier;

```

```

21         points_etablissement;
22         score}
23     → (Q.to_float points_licence,
24         Q.to_float points_bac,
25         Q.to_float points_diplomes,
26         Q.to_float points_etablissement,
27         Q.to_float points_dossier,
28         Q.to_float score))
29     in
30     remplissage y_semestres h_semestres 5 10;
31     remplissage y_baccalaureat h_baccalaureat 1 14;
32     remplissage y_autres_diplomes h_autres_diplomes 1 9;
33     if y_upec = 0. then h_upec.(0) ← h_upec.(0) + 1
34     else h_upec.(1) ← h_upec.(1) + 1;
35     remplissage y_appreciation h_appreciation 5 3;
36     remplissage y_total h_total 5 20;
37     (h_semestres, h_baccalaureat, h_autres_diplomes,
38      h_upec, h_appreciation, h_total)
39 )
40 (Array.make 11 0, Array.make 15 0, Array.make 10 0,
41  Array.make 2 0, Array.make 4 0, Array.make 21 0)
42 candidats_score
43 let histogramme_complet_semestre
44     (candidats_semestre : (string × dossier_formulaire) list)
45     : int array =
46 List.fold_left
47     (λ(h_semestres : int array)
48      → λ(_,c)
49       → let y_semestres
50          = (match score_semestres (notes_licence_of_candidat c)
51             ↪ with
52                {points_semestres}
53                → Q.to_float points_semestres)
54             in
55                remplissage y_semestres h_semestres 5 10;
56                h_semestres
57             )
58     (Array.make 11 0)
59     candidats_semestre

```

271 <LIASSE_NOMINAUX> ≡

OCaml

```

1 let print_histogramme_total oc h score =
2   Printf.fprintf oc
3   ↪ "\begin{tikzpicture}[x=0.0033\\textwidth,y=0.1\\baselineskip]\\n";
4   for i = 0 to 20
5   do
6     Printf.fprintf oc "\\draw[fill=black] (%d,0) rectangle
7     ↪ (%d,%d);\\n" (5*i) (5*(i+1)) h.(i)
8   done;
9   Printf.fprintf oc "\\node at (0,-3) {\tiny 0};\\n";
10  Printf.fprintf oc "\\node at (105,-3) {\tiny 105};\\n";

```

```

9   Printf.fprintf oc "\draw[->,color=gray,>=Latex] (%.2f,-10) to
   ↪ (%.2f,-0.5);\n" score score;
10  Printf.fprintf oc "\node at (%.2f,-12) {\tiny \num{%.2f}};\n"
   ↪ score score;
11  Printf.fprintf oc "\tikzset{baseline=(current bounding box)};\n";
12  Printf.fprintf oc "\end{tikzpicture}"

```

272 <LIASSENOMINAUX 271> +≡

OCaml

```

1   let print_histogramme ?(cut=0) oc h longueur score =
2   Printf.fprintf oc
   ↪ "{\begin{minipage}{0.5\textwidth}\begin{flushright}\n";
3   Printf.fprintf oc
   ↪ "\begin{tikzpicture}[x=0.0066\textwidth,y=0.01\baselineskip]\n";
4   for i = 0 to (Array.length h) - 1
5   do
6     Printf.fprintf oc "\draw[fill=%s] (%d,0) rectangle (%d,%d);\n"
7       (if longueur*(i+1) > cut then "black" else "gray")
8       (longueur*i) (longueur*(i+1)) h.(i)
9   done;
10  (if cut != 0
11  then Printf.fprintf oc "\draw[dashed,color=red] (%d,-50) to
   ↪ (%d,150);\n" cut cut);
12  Printf.fprintf oc "\node[anchor=east] at (-5,-30) {\tiny 0};\n";
13  Printf.fprintf oc "\node[anchor=west] at (%d,-30) {\tiny %d};\n"
14    (longueur × ((Array.length h) - 1) + 5) (longueur ×
   ↪ ((Array.length h)));
15  Printf.fprintf oc "\draw[->,color=gray,>=Latex] (%.2f,-100) to
   ↪ (%.2f,-5);\n" score score;
16  Printf.fprintf oc "\node at (%.2f,-120) {\tiny \num{%.2f}};\n"
   ↪ score score;
17  Printf.fprintf oc "\tikzset{baseline=(current bounding box)};\n";
18  Printf.fprintf oc
   ↪ "\end{tikzpicture}\end{flushright}\end{minipage}}"
```

273 <LIASSENOMINAUX 271> +≡

OCaml

```

1   let fiche oc (h_semestres, h_baccalaureat, h_autres_diplomes,
2     h_upec, h_appreciation, h_total) (numero,(c,cc,_)) =
3   match c,cc with
4     {nom;
5     prenom;
6     bac_plus_1 = L1(_,Droit(_),Some(s1),Some(s2),_);
7     bac_plus_2 = L2(_,Droit(_),Some(s3),Some(s4),_);
8     bac_plus_3 = L3(_,Droit(_),Some(s5),s6,_);
9     autre_licence;
10    baccalaureat;
11    upec;
12    bac_plus_4;
13    bac_plus_5;},
14   {ScoreCandidat.points_licence;
15   points_bac;

```

```

16     points_diplomes;
17     points_dossier;
18     points_etablissement;
19     score}
20 →
21     Printf.fprintf oc ⟨PRÉAMBULELIASSENOMINAUX 274⟩
22     Printf.fprintf oc ⟨LICENCELIASSENOMINAUXUN 275⟩
23     (match s6 with
24     | Some(s6') →
25         Printf.fprintf oc "&||textcolor{gray}{||textsc{s6:}}
26         ↪ ||num{%.2f}" s6'
27     | _ → ());
28     Printf.fprintf oc "\n\\end{tabular}\n";
29     Printf.fprintf oc "\\\hfill {\\Large score:
30     ↪ ||num{%.2f}||textcolor{gray}{/||num{55}}}\n"
31     (Q.to_float points_licence);
32     Printf.fprintf oc "|||||n|||||n";
33     print_histogramme ~cut:30 oc h_semestres 5 (Q.to_float
34     ↪ points_licence);
35     Printf.fprintf oc "\\bigskip\n\n";
36     Printf.fprintf oc ⟨INFORMATIONDOSSIERNOMINALBACCALAURÉAT 282⟩
37     print_histogramme oc h_baccalaureat 1 (Q.to_float points_bac);
38     Printf.fprintf oc "\\bigskip\n\n";
39     Printf.fprintf oc "\\noindent ||textcolor{gray}{\\Large Autres
40     ↪ diplômes nationaux}";
41     Printf.fprintf oc "\\\hfill {\\Large score:
42     ↪ ||num{%.0f}||textcolor{gray}{/||num{10}}}\n\n"
43     (Q.to_float points_diplomes);
44     Printf.fprintf oc "\\indent Licence: %s\n\n" (if autre_licence
45     ↪ then "oui" else "non");
46     Printf.fprintf oc "\\indent ||textsc{m}1: %s\n\n" (if
47     ↪ bac_plus_4 then "oui" else "non");
48     Printf.fprintf oc "\\indent ||textsc{m}2: %s\n\n" (if
49     ↪ bac_plus_5 then "oui" else "non");
50     print_histogramme oc h_autres_diplomes 1 (Q.to_float
51     ↪ points_diplomes);
52     Printf.fprintf oc "\\bigskip\n\n";
53     Printf.fprintf oc "\\noindent ||textcolor{gray}{\\Large Licence
54     ↪ préparée à l' ||textsc{upec}}";
55     Printf.fprintf oc "\\\hfill {\\Large score:
56     ↪ ||num{%.0f}||textcolor{gray}{/||num{5}}}\n\n"
57     (Q.to_float points_etablissement);
58     Printf.fprintf oc "\\indent %s\n\n" (if upec then "oui" else
59     ↪ "non");
60     Printf.fprintf oc "\\bigskip\n\n";
61     Printf.fprintf oc "\\noindent ||textcolor{gray}{\\Large
62     ↪ Appréciations}\n";
63     Printf.fprintf oc "\\\hfill {\\Large score:
64     ↪ ||num{%.1f}||textcolor{gray}{/||num{20}}}\n\n"
65     (Q.to_float points_dossier);

```

```

53     print_histogramme oc h_appreciation 5 (Q.to_float
      ↪ points_dossier);
54     Printf.fprintf oc "\\vfill";
55     Printf.fprintf oc "\\noindent {\\LARGE\\textcolor{gray}{Score
      ↪ total:} \\num{%.2f}\\textcolor{gray}{/\\num{105}}\\hfill"
56         (Q.to_float score);
57     print_histogramme_total oc h_total (Q.to_float score);
58     Printf.fprintf oc "\\clearpage\\n";
59 | _ → ()

```

274 <PRÉAMBULELIASSENOMINAUX> ≡ OCaml

```

1  "\\section{\\textsc{s}}\\n\\n" (String.capitalize_ascii
  ↪ (String.lowercase_ascii prenom)) (String.capitalize_ascii
  ↪ (String.lowercase_ascii nom));
2  "\\begin{flushright}\\textcolor{gray}{\\tiny
  ↪ \\textsc{s}}\\end{flushright}\\n\\n" (String.lowercase_ascii
  ↪ numero);

```

275 <LICENCELIASSENOMINAUXUN> ≡ OCaml

```

1  "\\noindent \\textcolor{gray}{\\Large Licence de droit dans une
  ↪ université publique française}\\n";
2  "\\|\\|";
3  "\\begin{tabular}{lll}\\n";
4  "\\textcolor{gray}{\\textsc{l}1}&\\textcolor{gray}{\\textsc{s1:}}
  ↪ \\num{%.2f}&\\textcolor{gray}{\\textsc{s2:}} \\num{%.2f}\\|\\|\\n"
  ↪ s1 s2;
5  "\\textcolor{gray}{\\textsc{l}2}&\\textcolor{gray}{\\textsc{s3:}}
  ↪ \\num{%.2f}&\\textcolor{gray}{\\textsc{s4:}} \\num{%.2f}\\|\\|\\n"
  ↪ s3 s4;
6  "\\textcolor{gray}{\\textsc{l}3}&\\textcolor{gray}{\\textsc{s5:}}
  ↪ \\num{%.2f}" s5;

```

276 <LIASSENOMINAUX 271> + ≡ OCaml

```

1  let liasse_nominaux nom liasse
2     (h_semestres, h_baccalaureat, h_autres_diplomes, h_upec,
  ↪ h_appreciation, h_total)
3     =
4     let oc = Stdlib.open_out ( <REPertoireTRAVAILSORTIES 188> ^ nom) in
5     Printf.fprintf oc "\\documentclass{lucart}\\n\\n";
6     Printf.fprintf oc "\\begin{document}\\n\\n";
7     Printf.fprintf oc "\\tableofcontents\\n\\n";
8     Printf.fprintf oc "\\clearpage\\n\\n";
9     List.iter (fiche oc (h_semestres, h_baccalaureat,
  ↪ h_autres_diplomes, h_upec, h_appreciation, h_total)) liasse;
10    Printf.fprintf oc "\\end{document}\\n\\n";
11    Stdlib.close_out oc

```

277 <PRODUCTIONNOMINAUX> ≡ OCaml

```

1 let () = liasse_nominaux "Liasse_nominaux.tex"
2     candidats_score_tri (histogrammes candidats_score_tri)

```

E.3 Fiches d'information finales

```

278 <FICHEINFORMATION> ≡ OCaml
1  let fiche_information
2      (numero,(c,rang) :
3          string × (dossier_formulaire × int option))
4      (candidats_score_tri :
5          (string × (dossier_formulaire × ScoreCandidat.t × float))
6          ⇨ list)
7      (tri_dossier : dossier_formulaire → groupes_formulaire)
8      (h_semestres, h_baccalaureat, h_autres_diplomes,
9          h_upec, h_appreciation, h_total :
10         int array × int array × int array
11         × int array × int array × int
12         ⇨ array)
13     (notes_licence_of_candidat : dossier_formulaire
14     → ScoreSemestres_in.t)
15 : unit
16 = let oc = Stdlib.open_out
17     ( <REPertoireTRAVAILSORTIES 188> ^
18     ⇨ "mail_information_candidats/"
19     ^ numero ^ ".tex") in
20 (match c with
21 {nom;
22 prenom}
23 →
24 Printf.fprintf oc <PréambuleFICHEINFORMATION 279>
25 (match (rang) with
26 | Some(i) →
27     (match (tri_dossier c) with
28     Nominal →
29     <INFORMATIONDOSSIERACCEPTÉNOMINAL 280>
30     | _ →
31     Printf.fprintf oc <INFORMATIONNonNOMINAL 286>
32     );
33     Printf.fprintf oc <INFORMATIONRANGDeCLASSEMENT 287>
34 | None →
35     (match (tri_dossier c) with
36     Nominal →
37     <INFORMATIONDOSSIERREFUSÉNOMINAL 288>
38     | _ →
39     Printf.fprintf oc <INFORMATIONNonNOMINAL 286>
40     );
41     Printf.fprintf oc <INFORMATIONREFUS 289>
42     )
43 );
44 Printf.fprintf oc <PAGEDEUXFICHEINFORMATION 290>
45 Printf.fprintf oc "\\end{document}\\n\\n";

```

43 **StdLib.close_out oc**

279 <PRÉAMBULEFICHEINFORMATION> ≡ OCaml

```

1  "\\documentclass{lucart}\n\n";
2  "\\begin{document}\n\n";
3  "\\begin{center}\n";
4  "\\textsc{université Paris xii (université Paris est
   ↪ Créteil)}\n\n";
5  "\\textsc{master Droit du numérique}\n\n";
6  "\\textit{Commission d'examen des candidatures au parcours
   ↪ Informatique et Droit}\n\n";
7  "\\textbf{Notice d'individuelle d'information}\n\n";
8  "\\end{center}\n";
9  "{\\noindent\\LARGE %s \\scshape %s}\n\n" (String.capitalize_ascii
   ↪ (String.lowercase_ascii prenom)) (String.capitalize_ascii
   ↪ (String.lowercase_ascii nom));
10 "\\bigskip\n\n";

```

280 <INFORMATIONDOSSIERACCEPTÉNOMINAL> ≡ OCaml

```

1  (match (List.assoc numero candidats_score_tri) with
2    {bac_plus_1 = L1(_,Droit(_),Some(s1),Some(s2),_);
3     bac_plus_2 = L2(_,Droit(_),Some(s3),Some(s4),_);
4     bac_plus_3 = L3(_,Droit(_),Some(s5),s6,_);
5     autre_licence;
6     baccalaureat;
7     upec;
8     bac_plus_4;
9     bac_plus_5};
10 {points_licence;
11  points_bac;
12  points_diplomes;
13  points_dossier;
14  points_etablissement},
15 score
16 →
17  Printf.fprintf oc <INFORMATIONDOSSIERNOMINALLICENCEUN 281>
18  (match s6 with
19    Some(s6') →
20    Printf.fprintf oc "&\\textcolor{gray}{\\textsc{s6:}}
   ↪ \\num{%.2f}" s6'
21    | _ → ());
22  Printf.fprintf oc "\n\\end{tabular}\n";
23  Printf.fprintf oc "\\vline\n";
24  Printf.fprintf oc "\\end{minipage}\\hfill";
25  print_histogramme ~cut:30 oc h_semestres 5 (Q.to_float
   ↪ points_licence);
26  Printf.fprintf oc "\\bigskip\n\n";
27  Printf.fprintf oc <INFORMATIONDOSSIERNOMINALBACCALAURÉAT 282>
28  print_histogramme oc h_baccalaureat 1 (Q.to_float points_bac);
29  Printf.fprintf oc "\\bigskip\n\n";

```

```

30 Printf.fprintf oc <INFORMATIONDOSSIERNOMINALAUTRESDIPLÔMES 283>
31 print_histogramme oc h_autres_diplomes 1 (Q.to_float
   ↪ points_diplomes);
32 Printf.fprintf oc "\\bigskip\\n\\n";
33 Printf.fprintf oc <INFORMATIONDOSSIERNOMINALUPEC 284>
34 Printf.fprintf oc "\\bigskip\\n\\n";
35 Printf.fprintf oc <INFORMATIONDOSSIERNOMINALAPPRÉCIATION 285>
36 print_histogramme oc h_appreciation 5 (Q.to_float
   ↪ points_dossier);
37 Printf.fprintf oc "\\vfill\\n\\n";
38 Printf.fprintf oc "{\\LARGE\\textcolor{gray}{Score total:}
   ↪ \\num{%.2f}\\textcolor{gray}{/\\num{105}}\\hfill"
39     score;
40 print_histogramme_total oc h_total score;
41 Printf.fprintf oc "\\vfill\\n\\n";
42 Printf.fprintf oc "Après en avoir délibéré,\\n";
43 | _ → ()
44 );

281 <INFORMATIONDOSSIERNOMINALLICENCEUN> ≡
  1 "\\noindent \\textcolor{gray}{\\Large Licence de droit dans une
   ↪ université publique française}\\n";
  2 "\\hfill {\\Large score:
   ↪ \\num{%.2f}\\textcolor{gray}{/\\num{55}}\\n" (Q.to_float
   ↪ points_licence);
  3 "||||";
  4 "\\begin{minipage}{0.3\\textwidth}\\n";
  5 "\\begin{tabular}{lll}\\n";
  6 "\\textcolor{gray}{\\textsc{l}1}&\\textcolor{gray}{\\textsc{s1:}}
   ↪ \\num{%.2f}&\\textcolor{gray}{\\textsc{s2:}} \\num{%.2f}||||\\n"
   ↪ s1 s2;
  7 "\\textcolor{gray}{\\textsc{l}2}&\\textcolor{gray}{\\textsc{s3:}}
   ↪ \\num{%.2f}&\\textcolor{gray}{\\textsc{s4:}} \\num{%.2f}||||\\n"
   ↪ s3 s4;
  8 "\\textcolor{gray}{\\textsc{l}3}&\\textcolor{gray}{\\textsc{s5:}}
   ↪ \\num{%.2f}" s5;

282 <INFORMATIONDOSSIERNOMINALBACCALAURÉAT> ≡
  1 "\\noindent \\textcolor{gray}{\\Large Baccalauréat}";
  2 "\\hfill {\\Large score:
   ↪ \\num{%.0f}\\textcolor{gray}{/\\num{15}}\\n\\n" (Q.to_float
   ↪ points_bac);
  3 "\\begin{minipage}{0.3\\textwidth}\\n";
  4 "\\indent %s |||" (string_of_baccalaureat baccalaureat);
  5 "\\end{minipage}\\hfill";

283 <INFORMATIONDOSSIERNOMINALAUTRESDIPLÔMES> ≡
  1 "\\noindent \\textcolor{gray}{\\Large Autres diplômes nationaux}";

```

```

2  "\\hfill {\\Large score:
   ↪ \\num{%.0f}\\textcolor{gray}{/\\num{10}}\\n\\n" (Q.to_float
   ↪ points_diplomes);
3  "\\begin{minipage}{0.3\\textwidth}\\n";
4  "\\indent Licence: %s\\n\\n" (if autre_licence then "oui" else "non");
5  "\\indent \\textsc{m}1: %s\\n\\n" (if bac_plus_4 then "oui" else
   ↪ "non");
6  "\\indent \\textsc{m}2: %s\\n\\n" (if bac_plus_5 then "oui" else
   ↪ "non");
7  "\\end{minipage}\\hfill";

```

284 <INFORMATIONDOSSIERNOMINALUPEC> ≡

```

1  "\\noindent \\textcolor{gray}{\\Large Licence préparée à
   ↪ l'\\textsc{upec}}";
2  "\\hfill {\\Large score:
   ↪ \\num{%.0f}\\textcolor{gray}{/\\num{5}}\\n\\n" (Q.to_float
   ↪ points_etablissement);
3  "\\indent %s\\n\\n" (if upec then "oui" else "non");

```

285 <INFORMATIONDOSSIERNOMINALAPPRÉCIATION> ≡

```

1  "\\noindent \\textcolor{gray}{\\Large Appréciations}\\n";
2  "\\hfill {\\Large score:
   ↪ \\num{%.1f}\\textcolor{gray}{/\\num{20}}\\n\\n" (Q.to_float
   ↪ points_dossier);
3  "\\begin{minipage}{0.3\\textwidth}\\n";
4  "\\(\\lqqquad\\)";
5  "\\end{minipage}\\hfill";

```

286 <INFORMATIONNONNOMINAL> ≡

```

1  "\\vfill\\n";
2  "Le processus d'examen des candidatures suivi par la commission, tel
   ↪ que prévu\\n";
3  "par le programme d'examen des candidatures, prévoyait de séparer
   ↪ les 725 candidatures\\n";
4  "en plusieurs groupes, permettant la comparaison entre les
   ↪ candidatures relevant de situations comparables.\\n";
5  "\\n\\n";
6  "Cela a amené à classer, dans un premier temps, 101 candidatures
   ↪ parmi celles ayant obtenu une licence de droit dans\\n";
7  "une université publique française en six semestres, ou sur le point
   ↪ de l'obtenir.\\n";
8  "\\n\\n";
9  "Votre candidature n'entrait pas dans ce cas, soit parce qu'elle
   ↪ présentait un\\n";
10 "parcours différent, soit du fait de la mauvaise qualité du
   ↪ remplissage du formulaire sur la plateforme nationale
   ↪ \\textsf{MonMaster}.\\n";
11 "\\n\\n";
12 "Elle a donc, dans un second temps, été\\n";

```

```

13 "évaluée de manière globale par la commission, et comparée avec les
    ↪ 101\n";
14 "candidatures déjà classées. À la suite de cette comparaison,\n";

```

```

287 (INFORMATIONRANGDECLASSEMENT) ≡
1  "la commission vous a classé:\n";
2  "\\bigskip\n\n";
3  "\\begin{center}\n";
4  "\\begin{tikzpicture}\n";
5  "\\node[scale=2] (rang) {Rang: \\HUGE %d};\n" i;
6  "\\node[above right=0.75\\baselineskip and 1cm of rang.east] {/17
    ↪ places};\n";
7  "\\node[right=of rang.east] {/134 classés};\n";
8  "\\node[below right=0.75\\baselineskip and 1cm of rang.east] {/725
    ↪ candidatures};\n";
9  "\\end{tikzpicture}\n";
10 "\\end{center}\n";

```

```

288 (INFORMATIONDOSSIERREFUSÉNOMINAL) ≡ OCaml
1  (match c with
2    {bac_plus_1 = L1(_,Droit(_),Some(s1),Some(s2),_);
3     bac_plus_2 = L2(_,Droit(_),Some(s3),Some(s4),_);
4     bac_plus_3 = L3(_,Droit(_),Some(s5),s6,_)}
5    ->
6    let points_licence = (match (score_semestres
    ↪ (notes_licence_of_candidat c))
7                          with {points_semestres}
8                              -> points_semestres)
9    in
10   Printf.fprintf oc (INFORMATIONDOSSIERNOMINALLICENCEUN 281)
11   (match s6 with
12     Some(s6') ->
13     Printf.fprintf oc "&\\textcolor{gray}{\\textsc{s6:}}
    ↪ \\num{%.2f}" s6'
14   | _ -> ());
15   Printf.fprintf oc "\n\\end{tabular}\n";
16   Printf.fprintf oc "\\vline\n";
17   Printf.fprintf oc "\\end{minipage}\\hfill";
18   print_histogramme ~cut:30 oc h_semestres 5 (Q.to_float
    ↪ points_licence);
19   Printf.fprintf oc "\\vfill\n\n";
20   Printf.fprintf oc "Du fait de vos résultats de licence,\n";
21 | _ -> ()
22 )

```

```

289 (INFORMATIONREFUS) ≡ OCaml
1  "et après avoir\n";
2  "vérifié que votre dossier ne présentait pas d'autres éléments\n";
3  "justifiant un traitement particulier,\n";
4  "la commission a décidé de ne pas\n";

```

5 "procéder au classement de votre candidature.\n\n";
 6 "\\bigskip\n\n";
 7 "La commission a évalué 725 candidatures pour 17 places et en a
 ↪ classé 134.";

290 <PAGEDEUXFICHEINFORMATION> ≡

1 "\\newpage\n\\vfill\\footnotesize\\setlength\\parindent{0pt}\n\n";
 2 "Le présent document est une notice individuelle d'information
 ↪ destinée à éclairer chacun·e des candidat·es\n\n";
 3 "au parcours Informatique \\& Droit du Master Droit du numérique de
 ↪ l'université Paris-Est Créteil sur la\n\n";
 4 "manière dont sa candidature a été examinée et évaluée en vue de son
 ↪ classement par la commission\n\n";
 5 "d'examen des candidatures. Il ne se substitue pas aux informations
 ↪ communiquées par la plateforme\n\n";
 6 "\\MonMaster{ }. \n\n";
 7 "\n\n";
 8 "Cette notice individuelle d'information est un document produit
 ↪ dans le cadre du programme conçu,\n\n";
 9 "paramétré et mis en oeuvre par La commission d'examen des
 ↪ candidatures du parcours Informatique \\& Droit\n\n";
 10 "pour assurer sa mission. Le programme en question, constitué d'un
 ↪ ensemble d'opérations automatisées et\n\n";
 11 "non automatisées, a pour finalité de rendre intelligible le
 ↪ processus d'examen, permettre une sélection de\n\n";
 12 "qualité fondée sur un traitement égalitaire des candidatures et
 ↪ assurer l'effectivité du droit au recours.\n\n";
 13 "\n\n";
 14 "Il est public, et mis à disposition à l'adresse suivante :
 ↪ \\url{https://lacl.fr/~lpellissier/master/admission.html}
 ↪ accompagné d'un guide d'explication. \n\n";
 15 "\n\n";
 16 "Ce programme a été conçu de manière à satisfaire le droit, garanti
 ↪ par l'article \\RR{311-3-1} du code des\n\n";
 17 "relations entre le public et l'administration, d'obtenir la
 ↪ communication des règles définissant le\n\n";
 18 "traitement et des principales caractéristiques de sa mise en
 ↪ oeuvre. Si vous estimez néanmoins que les\n\n";
 19 "informations communiquées demeurent insuffisantes, vous pouvez vous
 ↪ adresser à la présidente de\n\n";
 20 "l'université Paris est Créteil – 61 avenue du Général de Gaulle –
 ↪ 94010 Créteil Cedex. En l'absence de\n\n";
 21 "réponse dans un délai d'un mois à compter de la réception de votre
 ↪ demande par l'université, vous disposez\n\n";
 22 "d'un délai de deux mois pour saisir la Commission d'accès aux
 ↪ documents administratifs (\\textsc{cada}). Cette\n\n";
 23 "saisine constitue un préalable obligatoire à tout recours
 ↪ contentieux en matière de communication de\n\n";
 24 "documents administratifs. La saisine peut être formée : \n\n";
 25 "\\begin{itemize}\n\n";

26 *"\item sur papier libre, par lettre simple – 20 avenue de Ségur,*
 ↪ *75007 Paris ; \n";*

27 *"\item par voie électronique à l'adresse \texttt{cada@cada.fr}*
 ↪ *;\n";*

28 *"\item ou via un formulaire en ligne*
 ↪ *\url{https://www.cada.fr/formulaire-de-saisine}. \n";*

29 *"\end{itemize}\n";*

30 *"\n\pausepreamble";*

31 *"\n";*

32 *"La notice vous rappelle vos autres droits : \n";*

33 *"\begin{enumerate}";*

34 *"\item Le programme, dont ce document est un produit, a été conçu*
 ↪ *comme susceptible de faire l'objet de recours\n";*

35 *"en tant qu'il fixe les règles et modalités de l'ensemble des*
 ↪ *opérations automatisées et non automatisées\n";*

36 *"permettant l'examen et l'évaluation des candidatures au parcours*
 ↪ *Informatique et Droit. Dans un délai\n";*

37 *"de deux mois à compter de l'accomplissement des formalités*
 ↪ *adéquates de\n";*

38 *"publicité de ce programme, les recours suivants peuvent être*
 ↪ *introduits à l'encontre de ce\n";*

39 *"programme, conformément aux dispositions de l'article \RR{421-1}*
 ↪ *et suivants du code de\n";*

40 *"justice administrative et du livre \textsc{iv} du code des*
 ↪ *relations entre le public et l'administration : \n";*

41 *"\begin{itemize}";*

42 *"\item un recours gracieux adressé à la présidente de l'université*
 ↪ *Paris est Créteil – 61 avenue\n";*

43 *"du Général de Gaulle – 94010 Créteil Cedex ;\n";*

44 *"\item un recours contentieux en saisissant le tribunal*
 ↪ *administratif de Melun – 43, rue du\n";*

45 *"Général de Gaulle – Case postale no 8630 – 77008*
 ↪ *Melun Cedex ou par l'application\n";*

46 *"\url{https://www.telerecours.fr}.\n";*

47 *"\end{itemize}\n";*

48 *"\n";*

49 *"\item Les décisions individuelles relatives aux candidatures*
 ↪ *(classement ou refus), produites à\n";*

50 *"l'issue de l'ensemble des opérations automatisées et non*
 ↪ *automatisées prévues par le\n";*

51 *"programme, peuvent également faire l'objet de voies de recours*
 ↪ *distinctes. Dans un délai de\n";*

52 *"deux mois à compter de leur notification, ces décisions peuvent*
 ↪ *faire l'objet : \n";*

53 *"\begin{itemize}";*

54 *"\item d'un recours gracieux adressé à la présidente de*
 ↪ *l'université Paris est Créteil, dans les\n";*

55 *"mêmes conditions que ci-dessus ;\n";*

56 *"\item d'un recours contentieux en saisissant le tribunal*
 ↪ *administratif de Melun, dans les\n";*

57 *"mêmes conditions que ci-dessus.\n";*

58 `"\end{itemize}\n";`
59 `"\n";`
60 `"\item Dans la mesure où des données à caractère personnel vous`
61 `↪ concernant sont traitées dans le cadre du\n";`
62 `"programme, vous pouvez, par ailleurs, exercer vos droits d'accès,`
63 `↪ de rectification et de limitation sur\n";`
64 `"ces données, conformément aux dispositions des articles 15, 16 et`
65 `↪ 18 du règlement \textsc{ue} 2016/679 du Parlement\n";`
66 `"européen et du Conseil du 27 avril 2016 relatif à la protection des`
67 `↪ personnes physiques à l'égard du\n";`
68 `"traitement des données à caractère personnel et à la libre`
69 `↪ circulation de ces données, et abrogeant la\n";`
70 `"directive 95/46/\textsc{ce}, ainsi que votre droit d'opposition,`
71 `↪ conformément aux dispositions de l'article 21 du\n";`
72 `"même règlement. Les données traitées sont issues de la plateforme`
73 `↪ MonMaster, et le traitement repose sur\n";`
74 `"l'exécution d'une mission d'intérêt public (art. 6-1-e du`
75 `↪ règlement). Le responsable du traitement est\n";`
76 `"l'université Paris-Est Créteil, même s'il est concrètement mis en`
77 `↪ oeuvre par la commission d'examen des\n";`
78 `"candidatures, qui agit dans le cadre de ses attributions propres et`
79 `↪ dispose d'une autonomie dans\n";`
80 `"l'appréciation des candidatures et l'établissement du`
81 `↪ classement.\n\n Vos demandes doivent être adressées à la\n";`
82 `"déléguée à la protection des données de l'université Paris est`
83 `↪ Créteil, soit via l'adresse électronique :\n";`
84 `"\texttt{dpo@u-pec.fr}, soit par courrier postal à l'attention de`
85 `↪ cette déléguée à la protection des données, à la\n";`
86 `"même adresse que le courrier adressé à la présidente mentionné`
87 `↪ ci-dessus.\n\n Si, après avoir introduit ces\n";`
88 `"demandes, aucune réponse n'est apportée dans le délai d'un mois,`
89 `↪ éventuellement prolongé dans les\n";`
90 `"conditions prévues par les dispositions de l'article 12 §3`
91 `↪ règlement ou si cette suite n'est pas\n";`
92 `"considérée comme satisfaisante, une réclamation peut être adressée`
93 `↪ auprès de la \textsc{cnil} à l'adresse suivante :\n";`
94 `"3, place du Fontenoy – TSA 80715 – 75334 Paris Cedex 07. \n";`
95 `"\end{enumerate}\n";`
96 `"\pausepreambule";`
97 `"\n";`
98 `"Indépendamment de ces voies de recours, il vous est possible de`
99 `↪ prendre contact avec la commission\n";`
100 `"d'examen des candidatures, afin d'échanger à propos du programme et`
101 `↪ de sa mise en oeuvre.\n";`
102 `"Ces échanges, qui ne constituent pas des recours administratifs`
103 `↪ mais un simple canal de\n";`
104 `"communication, n'interrompent ni ne suspendent les délais de`
105 `↪ recours. Le courrier doit être\n";`
106 `"adressé au président de la commission d'examen des candidatures au`
107 `↪ parcours Informatique\n";`

- 86 "et Droit du Master Droit du numérique – \textsc{ufr} de droit –
↔ 83-85, avenue du Général de\textsc{n}";
- 87 "Gaulle – 94010 Créteil Cedex.\textsc{n}";

F Index

Les entrées sont numérotées par les numéros de paragraphes et non de pages. Un numéro est en gras si l'entrée y est définie, en italique si elle y est déclarée. Les entrées sont, le cas échéant, regroupées par bibliothèques. Une entrée d'une bibliothèque est en italique si elle lui est interne (non-exposée).

affiche_resultat, **I57**, I58
annee_universitaire, 23, **24I**
autres, **I49**, I50
autres.csv, **I49**, I50
autres_licence_droit, **I43**, I44, I54
autres_licence_droit.csv, **I43**, I44

but, **I46**, I47
but.csv, **I46**, I47

CalculDossiers, **I84**, I85, I93, I94
ajouter, **I52**, I53
appreciations_candidats, **7I**, 72
construit_candidat_final, 7I, **26I**
equivalences_baccalaureats, **I32**, I33
interclassement, **I53**, I54
notes_licence_of_candidat, 60, 68, **260**
tri_dossier, 24, 58, I30, I34, I38, I43, I46, I49, **253**

CalculScoresCasNominal, **I8I**, I84, I85, I9I–I94
CandidatCompletNominal, 77, **79**
appreciation_dossier, **67**, 26I
appreciation_lettre, **67**, 26I
baccalauréat, **83**
candidat_e_UPEC, **I16**, I25, 26I
licence_obtenue, **I07**, I23, 26I
master_années_validées, **I07**, I23, 26I
notes, **79**, 26I
CaracteristiquesBac, 23, **82**, 83, 84
Capacité, **I05**, I06, 245
GénéralSpécialités, **82**, 89, 245, 246, 254
GénéralSéries, **82**, 93, 245, 246, 254
Professionnel, **82**, 98, 246, 256
Technologique, **82**, 97, 246, 254
Équivalence, **I3I**, I35, 246, 256
ÉquivalenceÉvaluée, **I3I**, I32, I36, 245, 255
maximum_score_baccalauréat, 85, 89, **I04**, I27
maximum_score_notes_licence, 54, 56, **57**, I27
moyenne_pondérée, **49**, 52
NotesLicence, **47**, 48, 49, 79
PointsBacGénéral, 88, 89, **I0I**
points_spécialité, 88, 89, **I0I**
spécialité, 88, 89, I0I
PointsBacGénéralSérie, 92, 93, **I02**
points_série_général, 92, 93, **I02**

série, **92**, 93, 102
 PointsBacTechnologique, **96**, 97, 103
 points_série_technologique, 96, 97, **103**
 série, **96**, 97, 103
 score_autre_licence, 112, 115
 score_dossier_max, **76**, 127
 score_master1_max, 110, 113
 score_master2_max, 111, 114, 127
 score_max, 127
 score_établissement, 119, 120, 127
 ScoreAutreDiplômeNational, 108, **109–111**, 123
 années_validées_master, **108**, 110, 111, 123
 licence_obtenue, **108**, 112, 123
 points_diplômes, 108, **109–111**, 123
 ScoreBac, 84, **89**, **93**, **97**, **98**, **106**, 121, **135**, **136**
 baccalauréat, **84**, 89, 93, 97, 98, 106, 135, 136
 points_bac, 84, **89**, **93**, **97**, **98**, **106**, 121, **135**, **136**
 ScoreCandidat, 72, 77, 121–125, **126**, 183
 points_bac, 81, **121**, 126
 points_dossier, 81, **124**, 126
 points_établissement, 81, **125**, 126
 points_diplômes, 81, **123**, 126
 points_licence, 81, **122**, 126
 score, 77
 ScoreDossier, 74, **75**, 124
 appréciation_dossier, 74, 75, 124
 appréciation_lettre, 74, 75, 124
 points_dossier, 74, **75**, 124
 ScoreSemestres, **48**, **52**, 60, 61, 68, 122, 260
 points_semestres, 48, **52**, 61, 68, 122
 ScoreÉtablissement, 117, **118**, **119**, 125
 licence_UPEC, **117**, 119, 125
 points_etablissement, 117, **118**, **119**, 125
 SpécialitéGénéral, **86**, 87, 88, 242, 244
 Arts, **86**, 242
 BiologieÉcologie, **86**, 101, 244
 EducationPhysiquePratiquesCultureSportives, **86**
 HistoireGéographieGéopolitiqueEtSciencesPolitiques, **86**, 242
 HumanitésLittératurePhilosophie, **86**, 242
 LanguesLittératuresEtCulturesÉtrangères, **86**, 242
 LittératuresLanguesEtCulturesAntiquité, **86**
 Mathématiques, **86**, 101, 242, 244
 NumériqueEtSciencesInformatiques, **86**, 101, 242, 244
 PhysiqueChimie, **86**, 101, 242, 244
 SciencesDeLaVieEtDeLaTerre, **86**, 101, 242, 244
 SciencesDeL'ingénieur, **86**, 101, 244
 SciencesÉconomiquesEtSociales, **86**, 101, 242, 244
 SérieGénéral, **90**, 91, 92
 Littéraire, **90**, 102, 246
 Scientifique, **90**, 102, 245, 246
 ÉconomiqueEtSocial, **90**, 102, 245, 246

SérieTechnologique, **94**, 95, 96

- S2TMD, **94**
- ST2S, **94**, 246
- STAV, **94**
- STD2A, **94**
- STHR, **94**
- STI2D, **94**, 103, 246
- STL, **94**, 103, 246
- STMG, **94**, 103, 246

écartement, 52, **56**

candidats, **25**, 34, 38, **46**, 58, 130, **133**, 134, 138, 143, 146, 149, 158, 169

candidats_non, 155

candidats_score_tri, **72**, 73, 277

candidats_sous_seuil, **61**, 62, **66**, 154

candidats_sous_seuil.csv, **62**, **63**, 65

candidats_sur_seuil, **68**, 69, 72

candidats_non, **154**

candidats_tri, **73**, **154**, 155

Catholique, **238**, 240, 248

code de l'éducation

- article D. 334-1, 82
- article D. 336-1, 82
- article D. 336-3, 94
- article D. 337-51, 82, 99
- article D. 337-53, 99
- article D. 612-36-2-1, 7
- article D. 612-36-2-2-1, 5
- article L. 711-1, 28, 254, 256, 257

code des relations entre le public et l'administration

- article L. 300-4, 17
- article L. 311-3-1, 19
- article R. 311-3-1-1, 19
- article R. 311-3-1-2, 19

commission d'examen des candidatures, **1**, 2, 4, 5, 7, 11, 12, 20, 24, 30, 32, 36, 59, 63, 67, 70, 79, 98, 129, 137, 142, 145, 148, 178

- opérateur·ice du traitement, **8**, 37, 44, 168, 175, 178
- président·e, 9, 178
- rapporteur·ices, **10**, 63, 67, 129, 137, 142, 145, 148

corrections, **44**, 46

corrections.csv, **44**

corrige, **45**, 46

Csv, 192

- load, 200
- save, 203

csv (*Comma Separated Values*), 38, 39, 61, 68, 130, 138, 140, 143, 146, 149, 151, 152, 155, 169, 170, 199, 200

discipline, **234**, 235, 241

- Autre, **234**, 235, 250, 251
- Droit, **234**, 235, 250, 251, 254-261, 263, 273

dossier_formulaire, 21, **23**, 24, 33, 60, 140, 162, 183, 209

adresse_email, **23**, **38**, **169**, 231
 autre_licence, **23**, 231, 261
 bac_plus_1, **23**, 231, 261
 bac_plus_2, **23**, 231, 261
 bac_plus_3, **23**, 231, 261
 bac_plus_4, **23**, 231, 261
 bac_plus_5, **23**, 231, 261
 baccalaureat, **23**, 231, 261
 nom, **23**, 62, 69, 130, 138, 143, 146, 149, 157, 231, 263, 273
 prenom, **23**, 62, 69, 130, 138, 143, 146, 149, 157, 231, 263, 273
 upec, **23**, 231, 261

etablisement, **238**, 240, 241
 AutreEtablissement, **238**, 240
 Catholique, **238**
 UPEC, **238**, 240, 250
 établissement d'enseignement supérieur privé d'intérêt général (EESPIG), 142
 établissement public à caractère scientifique, culturel et professionnel (EPSCP),
 28

Float

compare, 72

groupes_formulaire, **20**, 24

Autre, **148**, 149
 AutreLicenceDroit, **142**, 143, 253
 BUTCarriereJuridiques, **145**, 146, 253
 Nominal, **28**, 38, 58, 169, 253
 NominalSaisieDefaillante, **137**, 138, 253, 257
 QuasiNominal, **128**, 130, 134, 253

Int

compare, 141

Lecture_ecriture, **182**, 183, 184, 185, 192–194

annee_master, 231, **251**
 annee_of_string, **233**, 248
 annees_universitaires, 231, **248**
 baccalaureat_of_type_caracteristiques, 231, **246**
 chargement_csv, **200**, 201, 207
 construction_candidat, **209**, 210
 construction_candidats, 21, 25, 44
 contains, **239**, 240
 discipline_of_string, **235**, 248
 ecriture_csv_travail, 38, 62, 69, 130, 138, 143, 146, 149, 158, 169, 202,
 203
 etablisement_of_string, **240**, 248
 fiche_information, 162, 163
 fiche_verification, 33, 34, **263**
 formulaire, **207**, 210
 lecture_csv_travail, 64, 65, 70, 133, 141, **201**
 lecture_liste_complementaire, 139, 140, **141**, 144, 147, 150

liasse_nominaux, 183
 meilleure_annee_de_licence, 231, **250**
 note_of_string, **237**, 248
 specialite_of_string, **242**, 246
 string_of_baccalaureat, **245**, 267, 273
 string_of_specialite, **244**, 245
 zip7, **247**, 248

List

assoc, 71, 132, 141, 152, 156
 filter, 38, 58, 61, 65, 68, 130, 134, 138, 143, 146, 149, 169
 fold_left, 66, 141, 215, 216, 223, 233, 249–251, 270
 fold_right, 248
 hd, 223
 length, 158
 map, 38, 62, 65, 69, 70, 72, 73, 130, 133, 138, 141, 143, 146, 149, 155, 158, 169,
 210, 233, 248
 mapi, 155
 mem, 68, 215, 216, 223
 nth, 215, 216, 223
 remove_assoc, 45, 66, 71, 132
 sort, 141, 158

loi Informatique et Libertés

article 47, 19

mail_information_candidats/candidats.csv, **169**

mail_verification_candidats/candidats.csv, **38**, 39, 170

Math, 51, **196**

pow, 56, 196, 197, **198**

base, 196

exposant, 196

message-mail, 42, 173

message-mail-user-agent, 37, 168

message-setup-hook, 37, 168

message-signature, 43

MonMaster, 2, 3, 5–9, 21, 22, 35, 36, 98, 158, 160, 164, 166, 167, 199, 204, 237

nominaux, **58**, 61, 68

nominaux_saisie_defaillante, **138**, 139, 154

nominaux_saisie_defaillante.csv, **138**, 139

note, **236**, 237, 241, 250

notes_appreciations, 70, 72

notes_licence_of_candidat, 61

ordre_resultat, **156**

PasEtablissement, **238**, 240, 248

PDF (*Portable Document File*), 17, 18, 35, 36, 61, 68, 130, 138, 143, 146, 149, 151,
 164, 166, 167, 204

Printf

fprintf, 263, 271–273, 276

Q, 191, 194

- of_float, 198, 260, 261
 - to_float, 61, 68, 72, 198, 273
- qpdf, 35, 164
- quasinominaux, **I30**
- quasinominaux.csv, **I30, I33**

- rattrapes, **65, 66, 68**
- Re**, 192
 - any, 223
 - bol, 223
 - compiles, 223, 239
 - compl, 223
 - digit, 223
 - eol, 223
 - exec, 239
 - matches, 223
 - opt, 223
 - rep, 223
 - seq, 223
 - set, 223
 - shortest, 223
 - str, 223, 239
- resultat, 163
- resultat_autres_apres, **I50**, 154
- resultat_autres_inter, **I50**, 154
- resultat_autres_licence_droit_apres, **I44**, 154
- resultat_autres_licence_droit_inter, 144, **I44**, 154
- resultat_autres_licence_droit_non, **I44**, 154
- resultat_autres_non, **I50**, 154
- resultat_but_apres, **I47**, 154
- resultat_but_inter, **I47**, 154
- resultat_but_non, **I47**, 154
- resultat_nominaux_saisie_defaillante_apres, **I39**, 154
- resultat_nominaux_saisie_defaillante_inter, **I39**, 154
- resultat_nominaux_saisie_defaillante_non, **I39**, 154
- resultat, **I55**, 158
- règlement général sur la protection des données
 - article 6, 36
 - article 15, 36
 - article 16, 36
 - article 18, 36
 - article 21, 36
 - article 22, 19

- seuil, **59**, 61, 68
- sortie.csv, 158
- Stdlib**
 - close_out, 263, 276
 - Obj**
 - repr, 198
 - open_out, 263, 276

String

- capitalize_ascii, 274
- compare, 158
- equal, 240, 248
- length, 245
- lowercase_ascii, 274
- split_on_char, 216, 223, 233, 246, 248
- starts_with, 235, 240, 248, 249

tri_dossier, 38, 169

Z, 191, 194

- of_int, 132, 261