

NOM Prénom :

CSC3102 – Contrôle Final 1

Année 2025 – 2026

Consignes :

- répondez aux questions en écrivant soit dans les cadres fournis au fil du sujet, soit dans les listings des scripts placés en fin d'exercice, aux endroits des marques correspondant aux questions ;
- si vous manquez de place, vous pouvez écrire en dernière page **en l'indiquant dans l'encart initial** ;
- les réponses qui ne requièrent pas de code doivent être justifiées ;
- photocopie, notes personnelles manuscrites et exemples de script sont autorisés ;
- la durée est fixée à 1h30 ;
- le barème est donné à titre indicatif.

Recherche dans les archives de la petite souris

Depuis l'apparition des premiers disques dur à la fin des années 50, la petite souris tient en registre numérique la date de chute de la première dent de chaque enfant. Les années passant, le nombre d'enfants devenant prohibitif, elle vous embauche comme stagiaire en système afin d'automatiser leur consultation.

[2 pts] Préambule

Le registre prend la forme d'un fichier ordinaire nommé *premieres_dents.txt*. Chaque ligne de ce fichier correspond à l'enregistrement de la chute de la dent d'un enfant ; construite sur le modèle suivant :

AAAA-JJ-MM;Nom;Prenom , AAAA pour l'année de la perte, JJ, le jour, MM, le mois.

Pour illustration, en voici un extrait de quelques lignes :

```
1985-28-02;Potter;Harry
1986-15-10;Granger;Hermione
1834-06-06;Elfe;Dobby
```

Le fichier *premieres_dents.txt* est placé dans le répertoire *Archives_petite_souris* situé à la racine du répertoire de connexion de la petite souris dont le login est *petite_souris*.

1) [1 pt] Étant connecté avec votre compte utilisateur, et ayant reçu les droits adéquats, donnez la commande qui permet de vous déplacer dans le répertoire *Archives_petite_souris*.

```
cd ~petite_souris/Archives_petite_souris
```

2) [1 pt] Précisez quels droits vous ont été octroyés pour que vous puissiez accéder à ce répertoire ainsi qu'aux fichiers qu'il contient.

```
Il faut avoir eu les droits de passage sur le répertoire de connexion de petite_souris,
les droits de passage et de lecture sur le répertoire Archives_petite_souris,
le droit de lecture sur les fichiers registres
```

[4,5 pts] Recherche initiale

D'ores et déjà placé(e) dans le répertoire *Archives_petite_souris*, construisez incrémentalement la suite de commandes qui permet de connaître l'année où un enfant a perdu sa première dent à partir de ses nom et prénom en procédant comme suit. Note pour notre cas d'étude : on suppose qu'il n'y a pas d'homonyme.

3) [1pt] Donnez la commande qui permet de chercher tous les enfants portant le nom stocké dans une variable *nom* pré-initialisée dans le fichier registre *premieres-dents.txt* du répertoire courant. De plus, afin d'éviter de sélectionner les noms qui englobent le nom recherché (*i.e.* pour ne pas garder un Dupont-Dupond quand on cherche un Dupont), affinez le motif de la recherche en vous appuyant sur le séparateur ; qui encadre strictement chaque information d'un enregistrement.

```
grep ";$nom;" premieres-dents.txt
```

4) [1pt] Complétez la commande pour filtrer votre recherche avec le prénom de l'enfant recherché, stocké dans une variable pré-initialisée *prenom*.

```
grep ";$nom;" premieres-dents.txt | grep "$prenom" ou grep "$nom $prenom" premieres_dents.txt
```

5) [1pt] La commande de la question précédente produit un affichage sur la sortie standard qui indique le fichier où les chaînes successivement recherchées ont été trouvées, insère une séquence "espace:espace", puis suit le formatage de l'enregistrement décrit au début du sujet. Par exemple, à ce stade, si on cherche quand Pierre Martin a perdu sa première dent, on obtient une sortie du type :

premieres-dents.txt : 2023-03-07;Pierre;Martin

Afin d'en tirer la date, procédez à plusieurs découpages successifs de la sortie de la commande obtenue en question (4). Pour ce faire, commencez par retirer la première partie de la sortie de la commande (4) pour ne conserver que l'enregistrement de la chute de la dent en utilisant l'espace comme séparateur ; puis gardez-en la date en sélectionnant le premier champ de la chaîne restante en prenant soin de préciser le bon séparateur ; pour finir , extrayez de nouveau le premier champ, en faisant aussi attention au séparateur adéquat, pour ne garder que l'année. Donnez la ligne de commandes correspondante.

```
grep ";$nom;" premieres-dents.txt | grep "$prenom" | cut -d ' ' -f 3 | cut -d ';' -f 1 | cut -d '-' -f 1
```

6) [1pt] Afin d'automatiser la recherche, écrivez un script *recherche_annee_perte_dent.sh* qui prend en paramètre les nom et prénom de l'enfant recherché ainsi que le chemin où trouver le registre. Après avoir vérifié que le nombre d'arguments est correct, le script doit avoir pour effet d'afficher l'année de perte de sa première dent grâce à la commande de la question (5). *Note : Le marqueur (11) sera complété lors de la question (11).*

recherche_annee_perte_dent.sh

```
#!/bin/bash
```

(11)

recherche_annee_perte_dent.sh (correction)

```
#!/bin/bash

if [ $# -neq 3 ] ; then
    echo « Nb de paramètres incorrect » >&2
    exit 1
fi

nom=$1
prenom=$2
registre=$3

#grep ";$nom;" premieres-dents.txt | grep "$prenom" | cut -d ' ' -f 3 | cut -d ';' -f 1 | cut -d '-' -f 1

(11)
annee=$(grep ";$nom;" premieres-dents.txt | grep "$prenom" | cut -d ' ' -f 3 | cut -d ';' -f 1 | cut -d '-' -f 1)
if [ -n "$annee" ] ; then
    kill USR1 $PPID
fi
```

7) [0,5pt] Donnez la commande qui rend exécutable le script *recherche_annee_perte_dent.sh*.

```
chmod u+x ./recherche_annee_perte_dent.sh
```

[1 pt] Réorganisation des archives

Avec les années, la taille du fichier *premieres-dent.txt* devenant prohibitive, et le temps de recherche associé avec, votre tuteur de stage vous demande d'améliorer le traitement. Votre première mesure consiste à fractionner ce fichier afin de l'annualiser en vue de réduire l'espace de recherche.

Pour l'exercice, on restreint la réorganisation aux enregistrements récoltés dans les années 2010.

8) [0,5pt] Écrivez **la** commande qui permet de rechercher toutes les lignes du fichier *premieres-dents.txt* qui sont préfixées par la chaîne « 2010- » et écrivez-les dans un fichier nommé *2010-premieres-dent.txt*.

```
grep "^2010-" premieres-dents.txt > 2010-premieres-dents.txt
```

9) [0,5pt] Systématisez cette réorganisation de 2010 à 2019 en écrivant une boucle qui itère de 0 à 9 et applique la commande précédente en s'appuyant sur la valeur de l'itérateur.

```
for i in 0 1 2 3 4 5 6 7 8 9 ; do # for i in $(seq 0 9); do
    grep "^201$i-" premieres-dents.txt > 201$i-premieres-dents.txt
done
```

[2 pts] Réduction de l'espace de recherche

L'évènement survenant généralement entre les 5 et 8 ans d'un enfant, vous écrivez à présent un script *recherche_ciblee_annee_premiere_dent.sh* qui permette de lancer la recherche uniquement sur les fichiers de registre adéquats. Vous avez déjà avancé avec un premier squelette que vous trouverez à la fin du sujet.

10) [1pt] Au repère (10), complétez le script *recherche_ciblee_annee_premiere_dent.sh* de manière à invoquer le script *recherche_annee_perte_dent.sh* écrit à la question (6) et que vous avez rendu exécutable en question (7).

11) [1pt] En l'état, les recherches sont systématiquement lancées sur les trois années cible, même si la recherche aboutit entre temps. Pour remédier à cela, dans le script *./recherche_annee_perte_dent.sh*, au repère (11), stockez le résultat de la recherche de l'année de la chute de la dent telle que réalisée à la question (6) dans une variable nommée *annee* ; puis si la recherche a abouti, *i.e.* si la variable *annee* n'est pas vide, envoyez un signal USR1 au processus parent, en l'occurrence *recherche_ciblee_annee_premiere_dent.sh* dans notre scénario. Si besoin, commentez les lignes devenues obsolètes en ajoutant un # en début de ligne.

recherche_ciblee_annee_premiere_dent.sh (correction)

```
#!/bin/bash

if [ $# -neq 3 ] ; then
    echo « Nb de paramètres incorrect » >&2
    exit 1
fi

(12) mkfifo resultat_recherche.pipe
    exec 3<>resultat_recherche.pipe

(13) trap 'read annee < resultat_recherche.pipe ; echo « Dent perdue en $annee »; exit 0' USR1

nom=$1
prenom=$2
annee_naissance=$3

debut_recherche=$(expr $3 + 5)
fin_recherche=$(expr $3 + 8)

while [ $debut_recherche -lt $fin_recherche ] ; do
(10)  ./recherche_annee_perte_dent.sh "$nom" "$prenom" "${debut_recherche}-premieres-dents.txt

    debut_recherche=$(expr $debut_recherche + 1)
done
```

[4 pts] Gestion du résultat de la recherche

Dans le script *recherche_annee_perte_dent.sh*, lorsque la recherche aboutit, le résultat est simplement écrit sur la sortie standard du processus. En vue de traiter ce résultat, vous décidez de mettre en place un mécanisme de communication afin de transmettre ce résultat au processus initiateur de la recherche.

12) [1pt] Dans le script *recherche_ciblee_annee_premiere_dent.sh*, mettez un place un tube nommé *resultat_recherche.pipe* ainsi que les redirections avancées nécessaires à son utilisation.

13) [1pt] Toujours dans le script *recherche_ciblee_annee_premiere_dent.sh*, mettez en place le gestionnaire de signal associé au signal émis à la question (11). Il doit avoir pour effet la lecture du résultat émis dans le

tube nommé *resultat_recherche.pipe*, l’affichage du message « Dent perdue en xx », xx correspond à ce que vous venez de lire, et mettre fin au processus pour couper court à la recherche dans d’autres fichiers.

Note : l’écriture du résultat de la recherche dans le tube nommé devrait avoir lieu dans le bloc de code que vous avez écrit pour répondre à la question (11). Même si cela serait nécessaire en pratique, pour des questions de lisibilité de la mise en page de l’examen, nous ne vous demandons pas de procéder à sa mise en place.

14) [2pts] Toujours en vue d’améliorer votre code, vous souhaitez lancer les recherches année par année en parallèle. Sans modification autre que la mise en parallèle des recherches lancées au repère (10), expliquez si et pourquoi le code reste correct ou non. Si le code reste correct, modifiez le script *recherche_ciblee_annee_premiere_dent.sh* en le rendant parallèle; sinon identifiez la section critique et protégez-la grâce aux scripts P.sh et V.sh utilisés en TP.

Les recherches peuvent être lancées en parallèle car il n’y a pas de donnée partagée et donc pas de concurrence.

[6,5 pts] Toutes leurs dents !

La capacité des disques durs ayant fortement augmentée, il est à présent possible d’enregistrer la date de chute de toutes les dents de tous les enfants !

Si on cherche non plus à connaître l’année de perte de la première mais l’historique de la chute de toutes les dents d’un enfant, les recherches doivent couvrir intégralement la plage d’années idoine. Le mécanisme à base de signaux qui interrompt la recherche à la première correspondance devient alors obsolète. Dans cet optique, vous rétablissez le code écrit à la question (6). Vous vous séparez également du tube nommé de la question (12) car vous décidez non plus de communiquer le résultat via ce dernier mais de produire un fichier historique.

15) [0,5pt] Donnez **la** commande qui permet de renommer le fichier du répertoire courant nommé 2010-premieres-dents.txt en 2010-dents.txt.

```
mv 2010-premieres-dents.txt 2010-dents.txt
```

16) [1pt] Dans le script *recherche_ciblee_multiple.sh* à la fin du sujet, version nettoyée de *recherche_ciblee_annee_premiere_dent.sh*, redirigez le résultat des appels répétés à *./recherche_perte_dent.sh* (appels que vous reportez de la question (10)) dans un fichier que vous nommez *Historique_[Nom]_[Prenom].txt*, Nom et Prenom devant être substitués par ceux de l’enfant faisant l’objet de la recherche. À cette étape, laissez les recherches se faire en séquentiel.

recherche_ciblee_multiple.sh (correction)

```
# !/bin/bash
```

```
if [ $# -neq 3 ] ; then
```

```
    echo « Nb de paramètres incorrect » >&2
```

```
    exit 1
```

```
fi
```

```

nom=$1
prenom=$2
annee_naissance=$3

debut_recherche=$(expr $3 + 5)
fin_recherche=$(expr $3 + 8)

while [ $debut_recherche -lt $fin_recherche ] ; do
(17) ./P.sh historique-$nom-$prenom.lock
(10)+(16)+(17)      ./recherche_perte_dent.sh  $1  $2  $debut_recherche-premieres-dents.txt  >>
Historique_$nom_$prenom.txt &
(17) ./V.sh historique-$nom-$prenom.lock

    debut_recherche=$(expr $debut_recherche + 1)
done

(18) wait
(18) echo $(wc -l Historique_$nom_$prenom.txt) dents perdues par $nom $prenom.

```

17) [2pts + 1pt bonus] À présent, **mettez en parallèle** la recherche année par année des différentes tombées de dents. Sans modification autre que la mise en parallèle des recherches, expliquez si et pourquoi le code reste correct ou non. Le cas échéant, identifiez la section critique et protégez-la grâce aux scripts *P.sh* et *V.sh* utilisés en TP.

Question Bonus : critiquer le mode de fonctionnement du code mis en place et expliquez comment il pourrait être amélioré.

L'ouverture du fichier historique est en accès concurrent.

(Pas terrible car le code devient séquentiel mais je m'arrête là quand même)

18) [2pts] Vous souhaitez rétablir le message de finalisation de la recherche. Pour cela, faites en sorte d'attendre la fin des recherches en parallèle avant d'afficher le message « xx dents perdues par Nom Prenom », xx étant le nombre de lignes du fichier historique constitué.

19) [1pt] Pour finir, vous souhaitez séparer les fichiers de données des scripts. Pour cela, donnez les commandes qui permettent de créer un répertoire *Registres*, d'y placer tous les fichiers suffixés par '.txt', de créer un répertoire *Scripts* et d'y placer tous les scripts (fichiers dont l'extension est '.sh'). On ne se soucie pas de la correction des chemins dans les scripts après cette réorganisation.

```

mkdir Registres
mv *.txt Registres
mkdir Scripts
mv *.sh Scripts

```

recherche_ciblee_annee_premiere_dent.sh

```
#!/bin/bash
```

```
if [ $# -neq 3 ] ; then
    echo « Nb de paramètres incorrect »
    exit 1
fi
```

(12)

(13)

```
nom=$1
prenom=$2
annee_naissance=$3
```

```
debut_recherche=$(expr $3 + 5)
fin_recherche=$(expr $3 + 8)
```

```
while [ $debut_recherche -lt $fin_recherche ] ; do
(10)
```

```
    debut_recherche=$(expr $debut_recherche + 1)
done
```

recherche_ciblee_multiple.sh

```
# !/bin/bash
```

```
if [ $# -neq 3 ] ; then
    echo « Nb de paramètres incorrect »
    exit 1
fi
```

```
nom=$1
prenom=$2
annee_naissance=$3
```

```
debut_recherche=$(expr $3 + 5)
fin_recherche=$(expr $3 + 8)
```

```
while [ $debut_recherche -lt $fin_recherche ] ; do
```

(10)+(16)+(17)

```
    debut_recherche=$(expr $debut_recherche + 1)
done
```

(18)