Polytech
Biotech III

Module : Bases de données et programmation.
Responsable : D. Puthier, Jacques van Helden & Nicolas Terrapon.
Date : Mardi 27 Janvier.
Durée : 2 heures.
Calculatrices/portables : non autorisés.
Documents : non-autorisés.
Nombre de pages : 4 (2x2)

Vous répondrez aux questions qui suivent sur la feuille d'examen qui vous sera fournie. Ecrivez distinctement. Si une instruction contient des espaces ou des tabulations, faites les apparaître clairement. Respectez bien les consignes qui vous sont données. N'oubliez pas d'indiquer vos noms et prénoms sur la feuille d'examen et de signer la feuille d'emargement après avoir rendu votre copie.

Le fichier "refGene_sub.txt", dont les premières lignes sont affichées ci-dessous, sera utilisé dans certains des exercices proposés. Ce fichier contient des informations proposées par la base de données refSeq que nous avons utilisée en cours.
Il s'agit d'un fichier tabulé à quatre colonnes dont chaque ligne contient les informations sur un transcrit humain. L'**identifiant du transcript** est donné en colonne 1. La colonne 2 donne le **symbole (nom) du gène** dont il est issu, la colonne 3 le **chromosome** qui le porte et la colonne quatre **la taille** de ce transcrit.

```
transcript      gene    chromosome      transcript_length
NM_000072       CD36    chr7    2104
NM_001001547    CD36    chr7    2065
NM_001127443    CD36    chr7    1810
NM_006532       ELL     chr19   4027
NM_001185074    ZCCHC6  chr9    4972
NM_002030       FPR3    chr19   2517
NM_001198673    TMEM136 chr11   4118
```

...

1 Premiers pas avec Unix (5 points)

Attention, vous devrez, toujours utiliser une commande (ou plusieurs) pour répondre à la question posée.
Mr L. Torwald est un étudiant de la promo Polytech 2015 qui fait ses premiers pas dans l'environnement Unix/Linux et qui s'est vu confié une analyse bioinformatique. Le chemin de son répertoire d'utilisateur (*home*) est :

```
/etudiants/2015/torwald
```

L. torwald s'est placé à la racine de l'arborescence, dans le répertoire "/" (c'est donc le répertoire "courant").

Proposez une instruction lui permettant :

1) De lister les fichiers et dossiers présents dans le répertoire courant.
2) De lister les fichiers et dossiers présents dans le répertoire courant en affichant les informations associées (propriétaire, groupe, droits,...).
3) De lister les fichiers et dossiers présents dans le répertoire courant y compris les fichiers cachés.
4) De lister, sans se déplacer, les fichiers présents dans son répertoire utilisateurs ("home").
5) De se déplacer dans son répertoire utilisateur.
6) De visualiser, avec la commande *less* le contenu du fichier "refGene_sub.txt" présent dans le répertoire "/tmp".
7) D'afficher les 10 premières lignes du fichier "refGene_sub.txt" présent dans le répertoire "/tmp".
8) D'afficher les 10 dernières lignes du fichier "refGene_sub.txt" présent dans le répertoire "/tmp".
9) D'extraire la colonne 3 du fichier "refGene_sub.txt" présent dans le répertoire "/tmp".
10) De rechercher, dans ce même fichier, les lignes commençant par "NM".
11) De rechercher, dans ce même fichier, les lignes commençant par "NM" et finissant par X ou Y.
12) D'afficher les 5 premières lignes de la colonne 3 du fichier "refGene_sub.txt" présent dans le répertoire "/tmp".
13) De copier le fichier "refGene_sub.txt" depuis le répertoire /tmp vers son répertoire utilisateur ("home").
14) De renommer le fichier "refGene_sub.txt" (répertoire utilisateur) en "refGene_sub_1.txt".

2 Zut ! (6 points)

L. Torwald débute... Il a par erreur utilisé la commande gshuf qui a randomisé les lignes de son programme *getMaxLen.py*. Ce programme permet, normalement, de **demander un nom de chromosome** à l'utilisateur et, après lecture du fichier *refGene_sub.txt* d'**afficher la taille du transcrit le plus long de ce chromosome**. On donne un exemple d'utilisation du programme dans le terminal :

```
$ python3 getMaxLen.py
Entrez une valeur pour le chromosome (1,2,3,...X,Y): X
37027
```

Question : réorganiser les lignes du programme pour que celui-ci puisse à nouveau faire ce pour quoi il a été écrit (vous pouvez omettre les commentaires mais votre copie doit contenir les lignes de code). Les lignes numérotées du programme sont données ci-dessous.

```
1              mx_l = int(l[3])
2      l = l.split("\t")
3  f = open("refGene_sub.txt", "r")
4  print(mx_l)
5      cc = l[2].lower() # le chromosome de la ligne courante
6  for l in f: # pour chaque ligne
7      l = l.rstrip("\n")
8      if cc == "chr" + ch:
9  ch = input("Entrez une valeur pour le chromosome (1,2,3,...X,Y): ").lower()
10 ch_l="1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,x,y"
11 ch_l = ch_l.split(",") # une liste contenant tous les chromosomes
12     exit()  # on quitte
13         if int(l[3]) > mx_l:
14 if ch not in ch_l: # si le chromosome n'est pas connu on quitte le programme
15 mx_l = 0
```

3 Où sont les commentaires ! (5 points)

Un collègue de L. Torwald, lui aussi débutant, lui a fourni du code dans un fichier au nom peu explicite : *script.py*. Le code ci-dessous n'est pas commenté mais on comprend vite que le script prend en entrée ce même fichier "refGene_sub.txt".

```
file_in = open("refGene_sub.txt", "r")
d = dict()

for l in file_in:

        l = l.rstrip("\n")
        l = l.split("\t")

        if l[1] in d:
                d[l[1]] = d[l[1]]+","+l[0]
        else:
                d[l[1]] = l[0]

for key in d.keys():
        print(key + "\t" + str(d[key]))
```

Questions 1 : recopiez le code sur votre copie en le commentant abondamment (3 points).
Questions 2 : qu'est censé faire ce programme ? (2 points)

4 Sélection aléatoire (4 points)

Ecrivez un programme permettant de lire le fichier "refGene_sub.txt" et de tirer 100 gènes au hasard parmi l'ensemble des gènes du chromosome 1 ('chr1'). Votre réponse devra comporter :

1. Le pseudo-code permettant de décrire la stratégie (2 points)
2. Le code du programme écrit en Python ? (2 points)

Vous devez dans ce programme utiliser la fonction randint() du module random. Celle-ci permet de tirer, au hasard, un entier dans un interval [n, m], On donne un exemple d'utilisation de la fonction randint().

```
from random import randint
n=2
m=5
randint(n,m) # un entier au hasard parmi: 2,3,4,5
```