

Travail à rendre

Travail à faire individuellement ou en binôme

- Pour chacune de vos commandes (scripts), les 3 points suivants doivent être respectés :
 - Votre code doit être commenté et chaque commande doit avoir un en-tête précisant son rôle et ses paramètres.
 - L'utilisateur doit pouvoir obtenir l'aide des commandes en tapant : `nomcommande --help`
 - Si l'utilisateur appelle la fonction avec de mauvais paramètres, un message d'erreur et d'aide approprié doit s'afficher.
 - A rendre sur Moodle ici : <https://cours.univ-grenoble-alpes.fr/mod/assign/view.php?id=302939>
au plus tard **le 29 novembre 2020 à 23h59**
- Attention ! Rendez votre TP avant la date limite. Il y aura 2 points de pénalité par jour de retard.**

Les questions suivantes concernent la gestion d'une épreuve de lancer de Javelot. Durant cette compétition, chaque athlète a eu 3 essais.

4.1 – Écrire la commande **maximum** qui calcule dans la variable nommée **maxi** et affiche le maximum d'une liste d'entiers passée en paramètres. Le nombre de paramètres n'est pas fixé.

Par exemple, la commande suivante : `maximum 1 2 3 4 3 -2 1 0 +4 2`

a pour effet d'afficher et de placer dans la variable **maxi** la valeur 4 car il s'agit du maximum des paramètres donnés dans l'exemple.

La commande **maximum** vérifiera la présence d'au moins un paramètre et affichera un message d'aide dans le cas contraire.

4.2 – Écrire la commande **categorie** qui calcule la catégorie d'âge d'un athlète dans une variable nommée **categ** et puis l'affiche. En cas d'erreur **categ** prendra la valeur "erreur" et de plus on affichera un message d'erreur sur la sortie erreur (**stderr**).

En athlétisme, les catégories d'âge de la saison 2020/2021 sont données par le tableau suivant :

CATEGORIE	ANNEE DE NAISSANCE	CATEGORIE	ANNEE DE NAISSANCE
Master	1986 et avant	Minime	2006 et 2007
Senior	1987 à 1998	Benjamin	2008 et 2009
Espoir	1999 à 2001	Poussin	2010 et 2011
Junior	2002 et 2003	Eveil Athlét	2012 à 2014
Cadet	2004 et 2005	Baby	2015 et 2016

Par exemple, la commande : `categorie 1990` a pour effet d'afficher la valeur Senior.

La commande `categorie 2017` a pour effet d'afficher la valeur erreur et le message "Trop jeune" sur la sortie erreur.

Pour afficher un message sur la sortie erreur on fait suivre le message de **>&2**

Exemple d'affichage sur la sortie **stderr** : `echo "Trop jeune" >&2`

4.3 – Écrire la commande **afficherresultat** qui affiche la catégorie et le meilleur lancé d'un athlète donné. Par exemple, la commande suivante : `afficherresultat Martin Galle 12-2-1991 64 47 53` a pour effet d'afficher : Le meilleur lancé du senior Martin Galle est de 64 m

La commande `afficherresultat` doit comporter 6 paramètres : le prénom et le nom de l'athlète, sa date de naissance (au format j-m-aaaa) et les 3 essais.

La commande `afficherresultat` pourra appeler les commandes `maximum` et `categorie`.

4.4 – On dispose d'un fichier de texte appelé `athletes.txt` et contenant une ligne pour chaque athlète avec le format suivant :

`Nom:Prenom:DatedeNaissance:essai1:essai2:essai3`

Exemple de fichier `athletes.txt` :

```
#Fichier de personnes
#Nom:Prenom:DateNaissance:essai1:essai2:essai3
Galle:Martin:30-11-2015:3:2:7
Pleur:Jean:13-3-1994:62:61:55
Prane:Dolly:19-11-1995:51:61:41
Mieu:Jeff-Edmond:22-12-2006:0:19:21
Honnete:Marie:24-9-2004:41:57:59
```

Ecrire la commande **repartirathletes** qui à partir du fichier `athletes.txt`, disperse les athlètes dans des fichiers différents en fonction de leur catégorie. **Ces fichiers seront nommés du nom de la catégorie** et de suffixe **.txt**. On créera donc les fichiers de nom `baby.txt`, `eveilathle.txt`, `poussin.txt`, `benjamin.txt`, `minime.txt`, `cadet.txt`, `junior.txt`, `espoir.txt`, `senior.txt`, `master.txt`

4.5 - Ecrire la commande **calculerresultats** qui affiche, pour chaque athlète d'un fichier donné, sa catégorie d'âge et son meilleur lancé. Le fichier contenant les athlètes et leurs performances au javelot est un paramètre de la commande.

Conseils :

1) Pensez à :

- Commenter votre code,
- Testez vos scripts séparément,
- Gérez les erreurs et les aides en ligne.

2) Si un script ne fonctionne pas, faites des traces en affichant les valeurs des variables mises en jeu. Ainsi vous trouverez plus facilement une erreur.

3) Attention : pour l'usage des caractères accentués dans les scripts, il est préférable d'utiliser les éditeurs **nano** ou **vim**, car ces caractères sont codés différemment dans les éditeurs `gedit` ou `scite`. Vous pouvez créer des scripts sans caractère accentué si vous le souhaitez.

4) Ne vous y prenez pas au dernier moment, commencez aujourd'hui même !

« Quoi que tu veuilles faire, fais-le rapidement. Ne remets pas à demain ce que tu peux faire aujourd'hui ! »

(Extrait d'un livre de sagesse tibétaine)

Compléments bash pouvant être utile

Tableaux

L'interpréteur bash permet d'utiliser des tableaux indicés

1. Création d'un tableau

La création d'un tableau indicé peut se faire simplement par initialisation, en fournissant ses éléments entre parenthèses :

```
$ t1=(12 18 5 -2 50)
$ t2=("un" "deux" "trois" "quatre")
```

Les indices sont assignés automatiquement, en commençant par 0.

Pour affecter un élément du tableau : `nomtableau[indice]=valeur`

Pour accéder à la valeur d'un élément du tableau on utilise la syntaxe `t${nomtableau[indice]}`

```
$ t2[1]="uno"
$ echo ${t2[1]}
uno
$ echo ${t2[0]}
un
```

Attention : si aucun indice n'est fourni pour le tableau, on accède à l'indice 0 :

```
$ t1=( 12 18 5 -2 50 )
$ echo $t1
12
```

Pour afficher l'ensemble du tableau on utilise la syntaxe `${nomtableau[@]}`

```
$ t1=( 12 18 5 -2 50 )
$ t1[5]=4
echo ${t1[@]}
12 18 5 -2 50 4
```

Lecture de fichier

Lecture de fichier dont les lignes sont formées de champs séparés par des ' : '
Voir le script **lectfic** donné en exemple.

Usage de fonctions

Voir dans le script **testnum** l'exemple de la fonction **isnum** qui détermine si l'argument passé en paramètre de la fonction est un entier.