DLST

UNIVERSITÉ Grenoble Alpes

INF203 - Partiel
Année 2021-2022

Durée : 1h30.
Une feuille A4 manuscrite recto/verso autorisée, dictionnaire papier (non annoté) autorisé pour les étudiants étrangers uniquement.
Tout autre document, calculatrices et appareils électroniques interdits.
Pour chaque question, une partie des points peut tenir compte de la présentation.
Le barème est indicatif.

Toute réponse même incomplète sera valorisée à partir du moment où elle réalise au moins une partie de ce qui est demandé. Les questions sont relativement indépendantes, et dans tous les cas vous pouvez utiliser les scripts et les fonctions demandées dans les questions précédentes même si vous n'avez pas réussi à les écrire.

Par ailleurs, tout code ou algorithme *élégant*, en plus d'être correct, sera bonifié.

1 Exercice issu de Caséine (3 pts)

1. (3 points) Écrivez un script qui affiche le nombre total de lignes contenant la chaine de caractères `trompe` dans les fichiers `us.txt` et `confidentiel.txt` (un seul nombre pour le total des deux fichiers).

Solution:

```
#!/bin/bash
N1=$(grep trompe us.txt | wc -l)
N2=$(grep trompe confidentiel.txt | wc -l)
expr $N1 + $N2
```

Ou, en plus élégant,

```
#!/bin/bash
cat us.txt confidentiel.txt | grep trompe us.txt | wc -l
```

2 Programmation en langage C (7 pts)

Le but de cet exercice est d'appliquer différentes opérations de mise en forme d'un titre pour l'afficher à l'écran.

Il vous est demandé d'écrire vous-mêmes tous les traitements demandés ; en particulier, il est interdit pour cet exercice d'utiliser les fonctions de la bibliothèque `string.h`.

Questions :

1. (2 points) Écrivez le code C de la fonction

   `void min2maj(char chaine[])`

   qui transforme en majuscules toutes les lettres minuscules de la chaîne reçue en argument. Les autres caractères doivent rester inchangés.

   Par exemple, si `chaine` contient `"Vive mon prof d'INF203 !!"`, après l'appel à `min2maj(chaine)`, elle contiendra `"VIVE MON PROF D'INF203 !!"`.

Solution: On se rappellera ici que `'A'-'a'=-32` :

```
void min2maj(char chaine[]) {
  int i=0;
  while (chaine[i] != '\0') {
```

```
    if (chaine[i] >= 'a' && chaine[i] <= 'z')
      chaine[i] -= 32;
    i++;
  }
}
```

2. (2 points) Écrivez le code C de la fonction

   `void encadre(char chaine[])`

   qui affiche à l'écran la chaîne reçue en argument, en l'encadrant par des étoiles. La largeur du cadre doit s'adapter à la longueur de la chaîne de caractères. On suppose que la chaîne s'écrit sur une seule ligne (qu'elle ne contient pas de caractère `'\n'`).
   Par exemple, si `chaine` contient `"Vive mon prof d'INF203 !!"`, on affichera :

```
****************************
* Vive mon prof d'INF203 !! *
****************************
```

Solution:

```
void encadre(char chaine[]) {
  int len=0;

  while (chaine[len] != '\0')
    len++;

  for (int i=0 ; i<len+4 ; i++)
    printf("*");
  printf("\n* %s *\n", chaine);
  for (int i=0 ; i<len+4 ; i++)
    printf("*");
  printf("\n");
}
```

3. (1 point) Même si vous n'êtes pas parvenu à écrire les fonctions précédentes, écrivez un programme principal qui déclare et initialise une chaîne de caractères (avec le contenu de votre choix), puis qui affiche cette chaîne encadrée et en majuscules (un seul affichage).

Solution:

```
int main(){
  char titre[80] = "Vive mon prof d'INF203 !!";

  min2maj(titre);
  encadre(titre);
  return 0;
}
```

4. (2 points) On souhaite modifier la fonction `encadre` pour que le cadre ait une largeur fixe de 80 caractères, avec le titre centré au milieu de la ligne.

   Comment calcule-t-on le nombre d'espaces à afficher à gauche et à droite du titre pour effectuer ce centrage? (il n'est **pas demandé** d'écrire complètement cette nouvelle fonction, à moins que vous ayez le temps)

   Par exemple, si `chaine` contient `"Vive mon prof d'INF203 !!"`, on veut afficher :

```
********************************************************************************
*                          Vive mon prof d'INF203 !!                           *
********************************************************************************
```

Solution: Il faut :
- compter la longueur du titre ;
- effectuer la différence entre la largeur du cadre et cette longueur ;
- diviser par 2 pour équilibrer les espaces à gauche et à droite.

En pratique on enlève encore 1 pour tenir compte des étoiles au bord du cadre, et on peut tenir compte de la parité de la longueur du titre..

```
void encadre_centre(char chaine[]) {
  int len=0;

  while (chaine[len] != '\0')
    len++;

  for (int i=0 ; i<80 ; i++)
    printf("*");
  printf("\n*");
  for (int i=0 ; i<(80-len)/2 - 1 ; i++)
    printf(" ");
  printf("%s", chaine);
  for (int i=0 ; i<(80-len)/2 - 1 + len%2 ; i++)
    printf(" ");
  printf("*\n");
  for (int i=0 ; i<80 ; i++)
    printf("*");
  printf("\n");
}
```

3 Redimensionnons nos photos ! (10pts)

Envoyer plus de trois photos dans le même email est devenu impossible aujourd'hui (et écologiquement peu responsable!). Nous allons dans cet exercice redimensionner toutes nos photos de vacances 2021 afin d'en faciliter l'envoi par email.

Nos photos sont contenues dans le répertoire ~`/Photos`, sous le format JPEG (extension `.JPG`). Dans ce répertoire, on trouve plusieurs sous-répertoires, comme par exemple les dossiers `Papa`, `LaFrangine`, `Moi`, etc., contenant les photos respectivement prises par Papa, la frangine et moi-même.

Questions :

1. (1 point) Si je me trouve dans mon répertoire principal (~), donnez la commande qui permet de lister le détail (droits, taille des fichiers, date d'accès, etc.) du contenu du dossier `Papa`. Vous redirigerez le résultat dans le fichier temporaire `/tmp/liste_photos`. La commande `cat /tmp/liste_photos` devrait vous donner typiquement un résultat de la forme suivante :

```
...
-rw-r--r-- 1 doisneau photographes  954752 juil 11 13:36 P1190386.JPG
-rw-r--r-- 1 doisneau photographes 2950144 juil 12 13:36 P1190387.JPG
-rw-r--r-- 1 doisneau photographes 2933248 aout  6 13:46 P1190388.JPG
-rw-r--r-- 1 doisneau photographes  956128 sept 21 13:56 P1190390.JPG
-rw-r--r-- 1 doisneau photographes 2398208 sept 24 14:10 P1190392.JPG
...
```

Solution: La solution de base est un simple :

```
ls -l ~/Photos/Papa > /tmp/liste_photos
```

On peut aussi faire mieux avec

```
ls -lrt ~/Photos/Papa > /tmp/liste_photos
```

pour lister les photos dans l'ordre chronologique de création (`t` pour le moment de création, `r` pour inverser l'affichage de plus ancien à plus récent).

2. (4 points) Nous souhaitons alors filtrer toutes les photos prises pendant le mois de juillet (indépendamment du photographe). En utilisant comme vous le voulez les commandes `tr`, `cut`, `grep`, `read`, etc., vues en cours, rédigez un script `filtrage_photos.sh` qui prend en argument un mois de l'année (sous le format à 4 lettres `janv`, `fevr`, ..., `dece`) et copie toutes les photos du mois donné en argument dans le répertoire ∼`/photos_xxxx`, où "`xxxx`" est remplacé par la chaîne de caractères représentant le mois (il faudra vérifier que le répertoire existe et le créer si ce n'est pas le cas). On supposera que le nom des photos lui-même ne contient jamais de chaînes de 4 caractères minuscules consécutifs.

Solution:

```
#!/bin/bash
if [ ! -e ~/photos_$1 ]
then
    mkdir ~/photos_$1
fi

cd ~/Photos
for i in *
do
    if [ -d "$i" ]
    then
        cd "$i"
        ls -lrt | grep "$1" | tr -s ' ' | cut -d ' ' -f 9 > /tmp/liste_noms_photos
        for fichier in $(cat /tmp/liste_noms_photos)
        do
            cp $fichier ~/photos_$1
        done
        cd ~/Photos ## Ne pas oublier de revenir dans le dossier initial
    fi
done
```

Alternativement et en plus élégant :

```
#!/bin/bash
if [ ! -e ~/photos_$1 ]
then
    mkdir ~/photos_$1
fi

cd ~/Photos
for i in *
do
    if [ -d "$i" ]
    then
    (
        cd "$i"
        ls -lrt | grep "$1" | tr -s ' ' | cut -d ' ' -f 9 |
        (
            while read fichier
            do
```

```
                cp $fichier ~/photos_$1
            done
        )
    )
    fi
done
```

3. (3 points) La commande `convert` de ImageMagick permet de modifier la résolution d'une image (et plein d'autres choses !) grâce à l'option `-resize geometry` où `geometry` est le format de conversion souhaité (par exemple `800x600` ou `640x480` pour des formats de photos 4 :3). Le format de la ligne de commande à utiliser est le suivant :

   `convert input-file [options] output-file`

   Créez un nouveau script `resolution_photos.sh` qui prend en arguments le nom d'un répertoire source ainsi que le format (`geometry`) de compression. Le script crée alors un sous-répertoire nommé `images_compressees` qui contiendra une copie des photos du répertoire source redimensionnées suivant le format donné en argument (ce sous-répertoire sera créé s'il n'existe pas). Le nom de ces fichiers redimensionnés sera au format `P1190392_640x480.JPG` si le fichier de départ était `P1190392.JPG`. Ce script prendra par défaut le format `640x480` s'il est exécuté avec un seul argument. Si le script est exécuté sans argument, un message d'erreur sera retourné.

Solution:

```
#!/bin/bash
cd $1

dest="images_compressees"
if [ ! -e $dest ]
then
    mkdir $dest
fi

resolution="640x480"

if [ $# -gt 1 ]
then
    resolution="$2"
fi

for i in *.JPG
do
    file=$(basename $i .JPG)
    convert $i -resize $resolution $dest/"$file"_$resolution.JPG
    ## Attention, "file_" est une autre variable !
    ## Il faut échapper avec "$file"_
done
```

4. (2 points) Grâce aux scripts précédents, écrire un dernier script interactif qui demande à l'utilisateur d'entrer un mois (sous format 4 lettres `xxxx`) ainsi qu'un format de conversion (par exemple `800x600`). Le script convertit automatiquement l'ensemble des photos prises par toute la famille pendant le mois et au format choisis et les place dans le dossier `~/photos_xxxx/images_compressees`.

Solution:

```
#!/bin/bash
echo "Choix du nom: \n"
read mois
echo "\n Choix du format: \n"
read format
```

```
./filtrage_photos.sh $mois
./resolution_photos ~/photos_$mois $format
```