Devoir Surveillé du 12 mars 2015 (corrigé)

Partie A

[*barême indicatif : 14 points*]

A1. Donnez le code de la fonction `readline` qui lit une ligne dans le fichier pointé par le descripteur `f`.

```
int readline(FILE* f,char ch[]){
  char c;
  int i=0;
  fscanf(f, "%c", &c);
  while (!feof(f) && c!='\n')
    {
      ch[i]=c;
      i++;
      fscanf(f, "%c", &c);
    }
  ch[i]='\0';
  return feof(f);
}
```

A2. Donnez le code de la fonction `versEntier` qui teste si la chaîne `ch` représente un entier positif ou nul et le cas échéant, stocke cet entier dans la variable dont l'adresse est dans `nb`.

```
int versEntier(char ch[],int *nb){
  int i=0;
  *nb=0;
  while(ch[i] != '\0') {
    if(ch[i]<'0' || ch[i]>'9')
      return 0;
    *nb = *nb * 10 + ch[i]-'0';
    i++;
  }
  return 1;
}
```

A3. Donnez le code de la fonction `trouve_colonne` qui renvoie l'indice de la `ligne` passée en paramètre (un tableau de caractères) où se trouve le premier caractère de la `i`ème colonne, en supposant que les colonnes sont séparées par le caractère `d`.

```
int trouve_colonne(char ligne[] ,char d,int i)
{
  int j=1,k=0;
  while(j!=i && ligne[k]!='\0'){
    if(ligne[k]==d)
      j++;
    k++;
  }
  if (j==i)
    return k;
  return -1;
}
```

Donnez le code de la fonction `extraction_colonne` qui copie dans la chaîne de caractères `col` le contenu de la $i^{ème}$ colonne de la `ligne` passée en paramètre, en supposant que les colonnes sont séparées par le caractère `d`.

```
int extraction_colonne(char ligne[],char col[], char d,int i){
  int j=0, deb=trouve_colonne(ligne,d,i);
  if(deb== -1) {
    col[0]='\0';
    return 0;
  }

  while(ligne[deb]!='\0' && ligne[deb]!=d){
    col[j]=ligne[deb];
    j++;
    deb++;
  }
  col[j]='\0';
  return 1;
}
```

A4. En utilisant les fonctions précédentes, donnez le code de la fonction `main` du programme `couper`. En particulier, vous devrez vérifier que tous les arguments fournis sont corrects.

```
int main(int argc, char* argv[]){
  char ligne[MAXTAILLE],colonne[MAXTAILLE];
  int num, c;
  FILE* f;

  if(argc != 4) {
    fprintf(stderr,"USAGE : %s delim col fichier\n",argv[0]);
    return 1;
  }
  if(strlen(argv[1]) > 1){
    fprintf(stderr,"le premier parametre doit etre un caractere\n");
    return 2;
  }
  if(!versEntier(argv[2],&num) || num <1){
    fprintf(stderr,"le second parametre doit etre un entier >= 1\n");
    return 3;
  }
  f=fopen(argv[3],"r");
  if (f==NULL) {
    fprintf(stderr,"Le fichier %s n'existe pas\n",argv[3]);
    return 4;
  }

  c=readline(f,ligne);
  while(!c){
    extraction_colonne(ligne,colonne,argv[1][0],num);
    printf("%s\n",colonne);
    c=readline(f,ligne);
  }

  /* Si derniere ligne non terminee par un saut de ligne */
  if (ligne[0] != '\0')
  {
    extraction_colonne(ligne,colonne,argv[1][0],num);
    printf("%s\n",colonne);
  }

  fclose(f);
  return 0;
}
```

A6. Écrivez un fichier ***Makefile*** pour la compilation de l'ensemble du programme.

```
couper: couper.o meschaines.o mesio.o conversion.o
        gcc couper.o meschaines.o mesio.o conversion.o -o couper
```

```
couper.o: couper.c meschaines.h mesio.h conversion.h
	gcc -c couper.c
meschaines.o: meschaines.c meschaines.h
	gcc -c meschaines.c
mesio.o: mesio.c mesio.h
	gcc -c mesio.c
conversion.o: conversion.c conversion.h
	gcc -c conversion.c
```

Partie B

[*barême indicatif : 7 points*]

B1. Écrivez un script qui prend deux noms de fichiers en arguments et qui affiche à l'écran :
- un message d'erreur si l'un des deux fichiers n'est pas accessible en lecture
- un message indiquant que les deux fichiers sont similaires s'ils ont le même nombre de lignes
- rien sinon

```
#!/bin/bash

if [ ! -r "$1" -o ! -r "$2" ]
then
	echo "L'un des deux fichiers $1 ou $2 n'est pas accessible en lecture"
else
	nb1=$(wc -l <$1)
	nb2=$(wc -l <$2)
	if [ $nb1 -eq $nb2 ]
	then
		echo "$1 et $2 sont similaires"
	fi
fi
```

B2. Écrivez un script qui prend deux noms de fichiers en arguments et qui affiche à l'écran :
- un message indiquant que le premier fichier est plagié sur le second si 90% ou plus de ses lignes sont présentes dans le second fichier
- rien sinon

```
#!/bin/bash

blocs_communs=$(similarites $1 $2)
commun=0
for bloc in $blocs_communs
do
	commun=$(expr $commun + $bloc)
done

total=$(wc -l <$1)
ratio=$(expr $commun \* 100 / $total)
if [ $ratio -ge 90 ]
then
	echo "$1 est un plagiat de $2 avec un ratio de $ratio%"
else
	echo "$1 n'est pas un plagiat de $2"
fi
```

B3. Écrivez un script qui prend deux noms de répertoires en arguments et qui pour tout couple (x, y) de noms de fichiers tel que x est un fichier contenu dans le premier répertoire et y un fichier contenu dans le second, appelle le script de la question 2 avec x et y en arguments.

```
#!/bin/bash

for source in $1/*
do
	for dest in $2/*
	do
		./plagiat.sh $source $dest
	done
done
```