Systèmes d'Exploitation && Programmation Concurrente

TRAVAUX PRATIQUES SEANCE 1

EXERCICES DE REPRISE

Exercice 1 - Arguments de main() – chaînes de caractères

❶ La signature complète de la fonction **main()** est : **int main(int argc, char *argv[]) ;** L'entier **argc** a pour valeur le nombre de paramètres reçus sur la commande de commande. **argv[0]** est une chaine de caractères contenant le nom de l'exécutable, **argv[1]** est une chaine de caractères contenant le premier argument, etc. Le tableau **argv** se termine par le pointeur **NULL** (c'est-à-dire **argv[argc]** contient le pointeur **NULL**).

Testez le programme **C** ci-dessous avec la ligne de commande suivante :
$ ./prog –c un deux 3

```
#include <stdio.h>
int main(int argc, char *argv[ ]) {
        int i ;
        printf("Nom du programme : %s\n" , argv[0]) ;
        printf("Nombre  d'arguments de la ligne de commandes : %d\n" , argc-1) ;
        printf("Les arguments sont : ") ;
        for(i=1 ; i<argc ; i++) {
                printf("\t%d : %s\n" , i , argv[i]) ;
        }
        return 0 ;
}
```

Terminologie :
Dans cet exemple, l'exécutable **prog** est lancé avec **4** arguments.
5 paramètres sont reçus dans la fonction **main()** à partir de la ligne de commande (les arguments + le nom du programme). **argv** est un tableau de taille **6** dont chaque entrée contient l'adresse d'une chaine de caractères, et dont la dernière entrée contient l'adresse **NULL**.

❷ Ecrivez un programme C (**miroir.c**) qui prend en paramètre une chaîne de caractères et l'affiche à l'envers. Par exemple :

$ ./miroir trace
```
> ecart
```

❸ Modifiez le programme **miroir.c** (en créant le programme **miroir_2.c**) pour qu'il puisse recevoir et traiter plusieurs arguments. Par exemple :

$ ./miroir ecart IRC
```
> trace CRI
```

Exercice 2 – utilisation des arguments de main()

Ecrire un programme **C** (**moyenne.c**) qui calcule et affiche la moyenne d'un ensemble de notes (nombres entiers) passées en arguments sur la ligne de commande. Le résultat doit être affiché sous la forme :

Votre moyenne est : résultat

La moyenne sera affichée tronquée à 2 décimales de précision.

Cahier des charges complet :

Vérifier que :

- Au moins une note est passée en argument. Si cette première vérification échoue, on affichera à l'écran : **Aucune moyenne à calculer**

- Chaque note est comprise entre **0** et **20** (bornes inclues). Si cette vérification échoue, on affichera à l'écran : **Note non valide**

- Si toutes les notes sont valides, on affichera sur la ligne de commande :
  Moyenne est : <valeur>
  <valeur> est la valeur de la moyenne tronquée à deux décimales.

Exemples de sorties attendues :

$ ./moyenne 10 15 15
```
> Moyenne est : 13.33
```

$ ./moyenne 8 7 12 15 3
```
> Moyenne est : 9.00
```

$ ./moyenne 8 maison 4
```
> Note non valide
```

$ ./moyenne 4 8 -1 7
```
> Note non valide
```

Note : le symbole **$** indique une entrée utilisateur sur la ligne de commande. Le symbole **>** indique un affichage du programme. Ce ne sont pas des caractères à afficher.

Aide :

- Avant de développer le code, réfléchissez aux type vous recevez vos arguments de la ligne de commande dans la fonction main. Quelle conversion est nécessaire ?
- Pour convertir une chaine de caractères représentant un nombre vers un type **integer** (**int**), il est possible d'utiliser les fonctions suivantes : **atoi**(), ou **sscanf**().
- Pour afficher **n** décimales d'un nombre flottant **x**, on pourra utiliser la syntaxe suivante :

printf("%1.nf", x) ; où **n** peut être remplacé par une valeur entière quelconque.

Exercice 3 - Manipulation des variables d'environnement

Ecrire un programme **C** (**VisuVariable.c**) permettant de visualiser sur l'écran la valeur d'une variable d'environnement donnée en paramètre (en ligne de commande). On utilisera la variable externe environ (**extern char **environ**). On aura besoin d'une fonction permettant de tester si une chaîne de caractères est ou non ***au début*** d'une autre chaîne [exemple : ''*Visu*'' est au début de ''*VisuVariable*''].

Quelques Indications :

- La commande shell **env** permet de visualiser les variables d'environnement avec leur valeur.
- La variable **environ** pointe sur la liste des variables d'environnement + valeurs.
- Utiliser la fonction **main**() avec paramètres : **int main(int nbarg , char **arg**). Le nom de la variable est passé en ligne de commande.