

## TP 2 - C

Les Tps se font sous un environnement linux, vous pouvez utiliser l'éditeur de texte que vous préférez.

### Partie I : Bases

#### 1. Moyenne

1. Ecrire un programme qui affiche la moyenne d'une suite d'entiers positifs entrés au clavier.

On arrêtera la saisie quand le nombre -1 est entré, comme dans l'exemple suivant.

Entrez un entier positif : 5

Entrez un entier positif : 2

Entrez un entier positif : 3

Entrez un entier positif : -1

La moyenne de ces 3 entiers vaut : 3.333333

#### 2. Min Max :

1. Créez un programme qui calcule le minimum et le maximum d'un tableau d'entiers de taille 5.
2. Plutôt qu'un tableau, faites entrer le nombre d'entiers par l'utilisateur puis demandez lui de les saisir.

#### 3. Des étoiles

1. Ecrire un programme qui affiche un triangle rempli d'étoiles, s'étendant sur un nombre de lignes entré au clavier, comme dans l'exemple suivant :

Nombre de lignes = 5

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

1. Demandez à l'utilisateur d'entrer une lettre ('g' ou 'd' pour droite ou gauche) permettant de changer le côté de la pointe du triangle (précédemment les étoiles étaient collées à gauche);
2. Demandez à l'utilisateur de choisir si le triangle doit pointer vers le haut ou vers le bas

#### 4. Tri à bulle

1. Créez un programme qui vérifie qu'un tableau d'entiers de longueur 6 est trié (les nombres sont dans l'ordre croissant).

2. Modifiez le programme pour qu'il inverse la position de tout élément plus grand que l'élément précédent avec ce dernier.

Par exemple pour le tableau 1 2 4 3 6 5 on inversera le 4 et le 3 puis le 6 et le 5.

1. Modifiez ce programme pour qu'il effectue cette action jusqu'à ce que le tableau soit trié

## 5. Comptage de lettres

1. Ecrire un programme qui compte le nombre de chacune des lettres de l'alphabet d'un texte écrit au clavier. Pour simplifier, on ne tiendra compte que des minuscules, mais on comptera le nombre de caractère non reconnus. La saisie sera de 100 caractères maximum.

## Partie II : Avancé

### 6. Mot de passe

On considère le code suivant qui demande un mot de passe à un utilisateur et lui propose de faire une requête à un administrateur si le mot de passe est faux :

```
#include "stdio.h"
#include "string.h"
#include "stdlib.h"

int main(){
    char message[200];
    char password[8] = "bonjour";
    char user_input[8];
    int logged_in = 0;

    while (!logged_in){
        // Ask the user for the password
        printf("please enter a password\n");
        scanf("%s", user_input);

        if (!strcmp(user_input, password)){
            logged_in = 1;
        } else {
            printf("Wrong password\nWould you like to send a request to the admin ? (y/n)");

            // ask the user if he wants to send a permission request to the admins
            char send_msg;
            scanf(" %c", &send_msg);
            if(send_msg == 'y'){
                printf("write your message here : \n");
                scanf("%s", message);
                printf(" we will send your request : (%s)\n", message);
            }

            // ask the user if they want to try again or leave
            printf(" do you wish to try again ? (y/n)");
            char try_again;
            scanf(" %c", &try_again);
            if (!try_again){
                exit(0);
            }
        }
    }
}
```

```
    }  
}  
printf("You are logged in ! \n");  
}
```

1. Ce code contient une faille de sécurité qui permet de s'identifier sans connaître le mot de passe, trouvez la et utilisez la pour arriver à faire afficher le texte "You are logged in !" sans taper le mot de passe.
2. Corrigez le code et vérifiez que la faille n'existe plus.