JUNIA Grande école d'ingénieurs

HEI · ISEN · ISA

CIR 2

POO/C++

TP N°1 : Rappel Langage C

Préalable :

Veuillez créer un dépôt Git pour l'ensemble de vos dépôt (GitHub, GitLab …). Pour votre travail de TD, vous pouvez par exemple vous baser sur les commandes cmake ci-dessous contenues dans un fichier CMakeLists.txt à placer à la racine de votre projet, et avec TD1, TD2… des répertoires.

/**********************/
project(TDs)

add_subdirectory(TD1)
add_subdirectory(TD2)
add_subdirectory(TD3)
add_subdirectory(TD4)
/**********************/
Dans chaque répertoire vous pourrez alors créer des fichiers CMakeLists.txt contenant par exemple des commandes « add_executable » pour le premier TD (TD1) :
/**********************/
add_executable(TD1_1 …)
add_executable(TD1_2 …)
…
/**********************/

Exercice 1 :

Compiler et exécuter le programme suivant sous l'IDE de votre choix et à l'aide de cmake, héberger vos sources sur votre dépôt.

```c
/* Premier programme en C */
#include<stdlib.h>
#include<stdio.h>

int main()
{
 printf("Hello World\n");
 return EXIT_SUCCESS;
}
```

Exercice 2 :

Ecrire le programme en C permettant le calcul des racines d'une équation du second degré ($ax^2+bx+c=0$).

Exercice 3 :

Ecrire une fonction « minuscule » qui retourne le caractère en minuscule d'un caractère donné en paramètre.

Exercice 4 :

Ecrire un programme permettant, via trois fonctions différentes, le calcul de « n! » :

- En itératif avec une boucle « for »
- En itératif avec une boucle « while »
- En récursif.

Exercice 5 :

Ecrire un programme qui permet de jouer au jeu du pendu :

- Le mot caché sera saisi par un autre joueur et stocké dans un tableau de type « char ».
- Après chaque proposition de lettre, on affiche de nouveau le mot en remplaçant les lettres cachées par un caractère spécial (« . », « _ », « * », ...), Par exemple: si le mot caché est « PROGRAMME » et que le joueur a trouvé le « P » et le « M », on affiche « P_____MM_ ».
- Si la lettre proposée n'appartient pas au mot caché, alors le nombre de tentatives restantes est décrémenté, dans la limite de 10 tentatives erronées possibles.
- Le jeu se termine lorsque le mot est trouvé ou quand le nombre de tentatives erronées est atteint.

Exercice 6 (tiré de quelques tests pour une formation CERFACS) **:**

Qu'imprime les programmes suivants :

```c
#include <stdio.h>
int main()
{
    int x = 14, y = 13;
    if (x > y)
    {
        if (x < 20)
        {
            x -= 10;
        }
        else
        {
            x += 10;
        }
    }
    printf("%d\n", x);
    return 0;
}
```

```c
#include <stdio.h>

int main()
{
    int x = 4, y = 3;

    x = x + y;
    y = x - y;
    printf("%d\n", y);
    return 0;
}
```

Combien de fois l'instruction « x += 1 » est exécutée ?

```c
int main()
{
    int x = 0, y = 100;

    while (x * x <= y)
    {
        x += 1;
    }
    return 0;
}
```

Exercice 7 :

Ecrire un programme qui permet de remplir aléatoirement un tableau de n valeurs entières et de les afficher par la suite dans un ordre croissant.

Exercice 8 :

Ecrire un programme qui permet d'allouer un tableau à deux dimensions n par m initialisé par des valeurs aléatoires qui seront triés dans un ordre croissant colonne par colonne (attention à l'agencement en mémoire des valeurs dans un tableau à deux dimensions).