

---

## TD 1

---

### Exercice 1.

```
#include <stdio.h>

int iX = 9;
int iJour, iMois, iAnnee;
const float dY = 7.0;

int main()
{
    int iX = 4;
    float fY;
    printf("La valeur de x est %d \n", iX);
    printf("Saisir une valeur pour y : ");
    scanf("%f", &fY);
    printf("La valeur de y est %f \n", fY);
    {
        int iX;
        float dY;
        printf("Saisir une valeur pour x : ");
        scanf("%d", &iX);
        printf("Saisir une valeur pour y : ");
        scanf("%f", &dY);
        printf("La valeur de x est %d \n", iX);
        printf("La valeur de y est %f \n", dY);
    }
    printf("Saisir une date suivant le format jour/mois/annee : ");
    scanf(" %d / %d / %d", &iJour, &iMois, &iAnnee);
    printf("La valeur de x est %d \n", iX);
    printf("La valeur de y est %f \n", dY);
    printf("Le jour saisi est %d \n", iJour);
    printf("Le mois saisi est %d \n", iMois);
    printf("L'annee saisie est %d \n", iAnnee);
    return 0;
}
```

1. Est ce que le programme compile avec la commande "gcc" sans option, il y a t-il des warnings, si oui lesquels.
2. lorsque l'on compile avec la commande "gcc -Wall -ansi -pedantic", y a-t-il des "warnings" supplémentaires ? si oui lesquels.
3. Lister les variables et les constantes globales de ce programme.
4. Combien y a-t-il de blocs d'instructions dans ce programme ? Pour chacun d'entre eux, énumérer leurs variables locales ainsi que les variables qui peuvent y être utilisées.
5. Où sont les déclarations et définitions des variables globales et locales.
6. Dérouler le programme et dessiner la pile.

**Exercice 2.** Soit l'algorithme suivant :

Deux variables entières en entrée :  $n_1$  et  $n_2$ .

Deux variables entières en sortie : *maximum* et *minimum*.

```
début
  lire ( $n_1$ )
  lire ( $n_2$ )
  écrire ("nombre 1 =",  $n_1$ , "nombre 2 =" ,  $n_2$ )
  si ( $n_1 \geq n_2$ ) alors
    maximum  $\leftarrow n_1$ 
    minimum  $\leftarrow n_2$ 
  sinon
    maximum  $\leftarrow n_2$ 
    minimum  $\leftarrow n_1$ 
  finsi
  écrire ("max(",  $n_1$ , ",",  $n_2$ , ") = ", maximum)
  écrire ("min(",  $n_1$ , ",",  $n_2$ , ") = ", minimum)
fin
```

1. Dérouler l'algorithme pour  $(n_1, n_2) = (5; 3)$  et  $(8, 8)$ .
2. Que fait-il ?
- c) Écrire le programme associé en langage C

**Exercice 3.**

Écrire deux fonctions **add** et **multipli** qui prennent respectivement en paramètre deux entiers et retourne respectivement leur somme et leur produit. Écrire un programme pour tester vos fonctions

**Exercice 4.**

Écrire une fonction qui prend en paramètre deux entiers et retourne la valeur minimale et la valeur maximale. Puis écrire un programme pour tester votre fonction.

Indication : vous pouvez utiliser l'algorithme de l'exercice 2

**Exercice 5.**

On voudrait avoir un programme qui demande à l'utilisateur d'indiquer s'il parle français, anglais, italien ou une autre langue et répond :

- "Salut !" s'il parle français,
- "Hello !" s'il parle anglais,
- "Ciao !" s'il parle italien,
- "Désolé je ne parle pas votre langue !" s'il ne parle ni français, ni anglais, ni italien.

**Questions**

- a) Analyser le problème à résoudre
- b) Écrire un algorithme.
- c) Écrire le programme associé en langage C

Indication : on pourra demander à l'utilisateur de répondre par F s'il parle français, A s'il parle anglais, I s'il parle italien ou par Z s'il parle une autre langue.