

# Corrigé du TP4

## Langage C

Y. ALJ

9 mars 2020

### 1 Corrigé des exercices

#### 1.1 Exercice 1 :

```
//exo1 question1
#include <stdio.h>

void main(){
    int i;
    for (i = 0; i<= 10; i++){
        printf("%d\n", i);
    }
}
```

```
//exo1 question2
#include <stdio.h>

void main(){
    int i;
    for (i = 100; i>= 0; i--){
        printf("%d\n", i);
    }
}
```

#### 1.2 Exercice 2 :

```
//exo2 question1
#include <stdio.h>

void main(){
    int i;
    for (i = 0; i<= 100; i++){
        if (i % 2 == 0){
            // on separe les nombres par un
            // espace
            printf("%d ", i);
        }
    }
}
```

```
//exo2 question2
#include <stdio.h>

void main(){
    int i;
    for (i = 0; i<= 100; i=i+2){
        printf("%d ", i);
    }
}
```

### 1.3 Exercice 3 :

```
//exo3 : nombres parfaits
#include <stdio.h>
void main(){
    int nombre;
    // variable pour calculer la somme des diviseurs de nombre
    int somme_div=0;
    int i;
    printf("Veuillez saisir un nombre\n");
    scanf("%d",&nombre);
    // on parcourt les nombres qui sont inferieurs a nombre
    for (i=1; i< nombre; i++){
        // on teste si le i courant est diviseur de nombre
        if (nombre % i == 0){
            // si oui on met a jour la variable somme_div
            somme_div += i; //pareil que somme_div = somme_div + i;
        }
    }
    // on teste ici si le nombre saisi est parfait on non
    if (somme_div == nombre){
        printf("Le nombre %d est parfait\n", nombre);
    }
    else{
        printf("Le nombre %d n est pas parfait\n", nombre);
    }
}
```

### 1.4 Exercice 4 :

```
//exo4.c calcul du pgcd
#include <stdio.h>

void main(){
    int nb1, nb2, i, pgcd;
    printf("Veuillez saisir un premier nombre\n");
    scanf("%d", &nb1);
    printf("Veuillez saisir un deuxieme nombre\n");
    scanf("%d", &nb2);

    for (i=1; i<= nb1 && i<= nb2; i++){
        // on cherche ici les diviseurs communs de nb1 et nb2
        if (( nb1 % i==0) && (nb2 % i ==0)){
            //ici on va trouver plusieurs valeurs
            // qui conviennent a notre condition
            // mais la derniere valeur trouvee sera la plus
            // grande et c est donc le pgcd
            pgcd = i;
        }
    }
    // on affiche le pgcd trouve
    printf("le pgcd de %d et de %d est %d", nb1, nb2, pgcd);
}
```

## 1.5 Exercice 5 :

```
#include <stdio.h>
//on inclut limits.h pour utiliser INT_MAX
#include <limits.h>

void main(){
    // on utilise le type long long pour pouvoir le resultat avec INT_MAX
    long long factorielle;
    int i;
    for (i=0; ; i++){
        if (i==0){factorielle = 1;}
        else{
            // on calcule la factorielle
            factorielle = factorielle * i;
            // si le nombre trouve est superieur a INT_MAX
            // on sort de la boucle
            if (factorielle > INT_MAX) {
                break;
            }
        }
        // on utilise le specificateur de format lli pour afficher
        // un entier de type long long
        printf("fact de %d est %lli\n", i, factorielle);
    }
}
```

## 1.6 Exercice 6 :

```
//exercice 6 table de multiplication
#include <stdio.h>

void main(){
    int i,j, N;
    printf("veuillez saisir un entier N\n");
    scanf("%d", &N);
    for (i =1; i<= N; i++){
        for(j = 1; j<=10; j++){
            // on utilise la tabulation \t pour mieux espacer les resultats
            printf("%d x %d = %d,\t", i, j, i*j);
        }
        printf("\n");
    }
}
```