TD 3.2

Exercice 1 (Diviseurs et nombres parfaits)
Voici un programme qui affiche la liste des diviseurs d'un entier naturel :.

```
#include <stdio.h>

int main ()
{
    int i,j;

    printf ("choisissez un nombre entier positif\n");
    scanf("%d",&i);
    while (i <= 0)
    {
        printf("il n'est pas positif ; choisissez-en un autre\n");
        scanf("%d",&i);
    }
    printf("voici ses diviseurs :\n");
    j = 1;
    while (j <= i)
    {
        if (i % j == 0)
            printf("\t%d\n",j);
        j = j + 1;
    }
    return 0;
}
```

1. *Faites le "tourner à la main" en supposant que le nombre choisi est* 10.
2. *Un nombre premier admet exactement deux diviseurs* (1 *et lui-même). Modifiez ce programme pour qu'il affiche en plus, selon le cas, le message "ce nombre est un nombre premier" ou "ce nombre n'est pas un nombre premier".*
3. *Un nombre parfait est la somme de tous ses diviseurs, lui-même exclu. Modifiez ce programme pour qu'il affiche en plus, selon le cas, le message "ce nombre est (respectivement n'est pas) un nombre parfait.*
4. *Modifiez ce programme pour qu'il affiche tous les nombres parfaits compris entre* 1 *et* 10 000.

Exercice 2 (pgcd)

1. *En s'inspirant du programme ci-dessus, écrire un programme qui détermine le pgcd de deux nombres a et b après les avoir saisis.*
2. *Ecrire un programme qui détermine pgcd(a, b) en utilisant l'algorithme d'Euclide.*
3. *Modifier les deux programmes qui déterminent pgcd(a, b) pour déterminer pgcd(a, b, c) (voir le rappel ci-dessous).*

Rappel sur l'algorithme d'Euclide

L'algorithme d'Euclide permet de déterminer le plus grand commun diviseur de deux entiers. Si $A \geq B > 0$, pour trouver pgcd(A,B) on fait la division euclidienne de A par B, on remplace B par le reste de cette division et A par B. On répète ces manipulations jusqu'à ce qu'on trouve un reste nul. Le dernier reste non nul est le pgcd de A et B.