

## Exercice 1

Écrire un programme qui affiche un tableau pour convertir les degrés Fahrenheit en degrés Celsius. Par exemple, votre programme affichera sur la colonne de gauche les nombre de 0 à 100 (de 5 en 5) correspondant aux degrés Fahrenheit, et sur la colonne de droite les valeur correspondantes en Celcius.

Rappel :  $c = 5 \times (f - 32)/9$

## Exercice 2

1. Tracer l'exécution du programme suivant :

```
#include <stdio.h>
main(){

    const int N=10;
    int i;
    int som;
    float harmo;

    harmo = 0.0; som = 0; i = 1; /* initialisation des variables */
    //Point d'observation 1

    while (i<=N){
        som = som + i;
        harmo = harmo + 1/(float)i;
        i++;
        // Point d'observation 2
    }
    // Point d'observation 3

    printf("La somme des %d premiers entiers est : %d\n", N, som);
    printf("La somme des %d premiers termes de la"
           " serie harmonique vaut : %f\n", N,harmo);
}
```

## Exercice 3

On rappelle que la suite de Fibonacci est définie par :  $f_1 = f_2 = 1$  et  $\forall n \geq 2, f_n = f_{n-1} + f_{n-2}$ . On pose pour tout  $n \geq 2, u_n = f_{n+1}f_{n-1} - f_n^2$ . Écrire un programme qui calcule  $u_n$  et affiche sa valeur pour  $n = 2, \dots, a$  où  $a$  est un nombre entré par l'utilisateur.

## Exercice 4

1. Écrire un programme qui demande 2 entiers  $x, y$  à l'utilisateur et qui affiche en retour un rectangle de  $x$  lignes et  $y$  colonnes. Par exemple, si l'utilisateur choisit  $x = 5, y = 3$  le programme devra afficher :

```
xxx
xxx
xxx
xxx
xxx
```

2. modifier le programme précédent. Cette fois, un seul nombre  $x$  est demandé et le programme affiche un triangle de  $x$  lignes comme suit :

```
x
xx
xxx
xxxx
xxxxx
```

3. Dernière modification, le programme doit afficher le triangle ci-dessous pour  $x = 5$  :

```
  x
   xx
    xxx
     xxxx
      xxxxxx
       xxxxxxxx
```

## Exercice 5

1. Programmer l'algorithme d'Euclide vu en cours (pour calculer le PGCD). Le programme doit en plus afficher le nombre de soustractions nécessaires pour parvenir au résultat.

2. On propose l'algorithme suivant pour le calcul du pgcd de  $a$  et  $b$ .

$$r_0 = |a|$$

$$r_1 = |b|$$

Pour tout  $n$ , si  $r_{n-1} \neq 0$ , alors on définit  $r_n$  par :

$$r_n = |r_{n-2} - r_{n-1}|$$

On peut montrer que le dernier terme non nul de la suite  $(r_n)$  est le PGCD de  $a$  et  $b$ .

Programmer cet algorithme. Le programme doit en plus afficher le nombre de soustractions nécessaires pour parvenir au résultat.

3. Comparer les deux algorithmes en terme de nombre d'itérations dans la boucle.