

- Écrivez **deux** programmes qui lisent un entier **n** saisi par l'utilisateur et affichent "Bonjour!" **n** fois à l'écran grâce à :
 - Une boucle *while* pour le premier programme,
 - Une boucle *for* pour le second.
- Soit S_n la somme des n premiers entiers positifs :

$$S_n = \sum_{i=1}^n i = 1 + 2 + \dots + (n - 1) + n$$

- Écrivez **deux** programmes qui lisent un entier **positif** n saisi par l'utilisateur, calculent **naïvement** S_n (grâce à l'équation ci-dessus) puis l'affichent.
- Le premier programme **doit** utiliser une boucle *while*,
 - Le deuxième **doit** utiliser une boucle *for*.
- Écrivez un programme qui lit deux entiers positifs **a** et **b** saisis par l'utilisateur. Calculez la valeur de $\mathbf{a} \times \mathbf{b}$, sans vous servir de l'opérateur de multiplication $*$.
 - Implémentez un programme qui, dans la fonction **main**, affiche la table ASCII (pour des valeurs de 0 à 127), sachant que :
 - Les caractères imprimables sont ceux dont la valeur décimale est dans [32, 126].
 - Les caractères imprimables doivent être affichés en tant que tels,
 - Les caractères non imprimables doivent être affichés par "XXX".

Voici l'affichage que vous devez obtenir :

```

caractère : XXX, décimal : 0, héxa : 00
...
caractère : ' ', décimal : 32, héxa : 20
...
caractère : 'd', décimal : 100, héxa : 64
caractère : 'e', décimal : 101, héxa : 65
...
caractère : XXX, décimal : 127, héxa : 7F

```

- On suppose qu'il existe un entier positif **value** dans le programme. Écrivez un extrait de code qui affiche **value** grâce à nombre d'étoiles sur une ligne (plutôt que directement afficher la valeur numérique). On affichera une étoile par tranche de 10 franchie par **value** :
 - on affichera "****\n" pour **value**=36,
 - on affichera "*****\n" pour **value**=128...

6. Écrivez un programme qui affiche une table de multiplication (de 0 à 9, en base 10), grâce à des boucles *for*. Résultat attendu :

```

0 0 0 0 0 0 0 0 0 0
0 1 2 3 4 5 6 7 8 9
0 2 4 6 8 10 12 14 16 18
0 3 6 9 12 15 18 21 24 27
0 4 8 12 16 20 24 28 32 36
0 5 10 15 20 25 30 35 40 45
0 6 12 18 24 30 36 42 48 54
0 7 14 21 28 35 42 49 56 63
0 8 16 24 32 40 48 56 64 72
0 9 18 27 36 45 54 63 72 81

```

7. On suppose qu'il existe un entier positif **value** dans le programme. Écrivez un extrait de code qui affiche une pyramide vers le haut à **value** lignes. Par exemple, si **value** = 3, on affiche la pyramide suivante :

```

*
***
*****

```

8. On suppose qu'il existe un entier positif **value** dans le programme. Écrivez un extrait de code qui affiche une pyramide vers la droite de **value** colonnes. Par exemple, si **value** = 3, on affiche la pyramide suivante :

```

*
**
***
**
*

```

9. Écrivez un programme qui affiche la table des différences de 0 à 9, en base 10. Résultat attendu :

```

x| 0  1  2  3  4  5  6  7  8  9
--+-+-----
0| 0  1  2  3  4  5  6  7  8  9
1|-1  0  1  2  3  4  5  6  7  8
2|-2 -1  0  1  2  3  4  5  6  7
3|-3 -2 -1  0  1  2  3  4  5  6
4|-4 -3 -2 -1  0  1  2  3  4  5
5|-5 -4 -3 -2 -1  0  1  2  3  4
6|-6 -5 -4 -3 -2 -1  0  1  2  3
7|-7 -6 -5 -4 -3 -2 -1  0  1  2
8|-8 -7 -6 -5 -4 -3 -2 -1  0  1
9|-9 -8 -7 -6 -5 -4 -3 -2 -1  0

```