TD #2 – DEV

19 septembre 2023
Abdel Bourjij
Erwan Kerrien

1 Boucles

1. Écrire un algorithme qui affiche un rectangle d'étoiles "*". La largeur et la hauteur seront entrées par l'utilisateur.
2. Écrire un algorithme qui affiche un triangle isocèle formé d'étoiles. La hauteur (nombre de colonnes) sera donnée en entrée et le triangle pointera vers la droite. Par exemple, pour une hauteur indiquée de 4, le résultat sera :

```
*
**
***
****
***
**
*
```

   — Modifier l'algorithme précédent pour que la pointe soit en haut. Ici, la hauteur est le nombre de lignes. Par exemple, pour une hauteur indiquée de 4, le résultat sera :

```
   *
  ***
 *****
*******
```

   — Modifier l'algorithme précédent pour que triangle soit empli de 'o' et entouré de chiffres indiquant le numéro de la ligne. Par exemple, pour une hauteur indiquée de 4, le résultat sera :

```
   1
  2o2
 3ooo3
4444444
```

2 Manipulation de réels

1. Écrire un algorithme qui résout une équation du second degré et affiche le résultat. Les paramètres a, b et c seront entrés par l'utilisateur de manière à résoudre l'équation $ax^2 + bx + c = 0$.
2. Soit la fonction $f(t) = 2t^3 - t^2 - 37t + 36$. Écrire un algorithme qui affiche la valeur minimale et la valeur maximale prise par cette fonction sur l'intervalle $[-5, 5]$. Pour ce faire, on calculera toutes les valeurs prises par la fonction pour chaque valeur de t allant de -5 à 5, tous les 0.25, et on en déterminera le minimum et le maximum.
3. Écrire un algorithme qui calcule et affiche la racine carré de chaque nombre x entré par l'utilisateur. Pour ce faire :
   — les nombres négatifs seront refusés (un message d'erreur s'affichera)
   — le calcul et l'affichage de la racine carré est effectué à chaque fois qu'un nombre est entré et tant que le nombre entré n'est pas 0 (zéro).
   — le calcul ne fera pas appel à la fonction racine carrée, mais suivra la méthode de Héron (aussi appelé méthode babylonienne) qui calcule les termes successifs de la suite

$$a_0 = x \qquad (1)$$
$$a_{n+1} = \frac{1}{2}\left(a_n + \frac{x}{a_n}\right) \qquad \forall n \geq 0 \qquad (2)$$

   — Cette suite converge vers la racine carré de x :

   — dans un premier temps, vous considérerez un nombre prédéfini de 100 itérations
   — dans un deuxième temps, on fera en sorte que le résultat soit précis jusqu'à la deuxième décimale, sans prédéfinir le nombre d'itérations à effectuer.

4. Écrire un algorithme qui demande à l'utilisateur d'entrer un nombre n et affiche le résultat de la somme

$$S_1 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots + \frac{1}{n}$$

   — Modifier l'algorithme précédent pour qu'il affiche aussi le résultat de la somme

$$S_2 = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \cdots \pm \frac{1}{n}$$