

TD2

EXERCICE 1: Etoiles

Écrire un programme qui affiche à l'écran 10 étoiles sous la forme suivante:

```
    *
   *
  *
 *
etc....
```

EXERCICE 2: Table de multiplication

Écrire un programme qui affiche la table de multiplication des entiers inférieur ou égaux à 12. Votre affichage doit être le suivant:

	1	2	3	4	5	6	7	8	9	10	11	12
1 :	1	2	3	4	5	6	7	8	9	10	11	12
2 :	2	4	6	8	10	12	14	16	18	20	22	24
3 :	3	6	9	12	15	18	21	24	27	30	33	36
4 :	4	8	12	16	20	24	28	32	36	40	44	48
5 :	5	10	15	20	25	30	35	40	45	50	55	60
6 :	6	12	18	24	30	36	42	48	54	60	66	72
7 :	7	14	21	28	35	42	49	56	63	70	77	84
8 :	8	16	24	32	40	48	56	64	72	80	88	96
9 :	9	18	27	36	45	54	63	72	81	90	99	108
10 :	10	20	30	40	50	60	70	80	90	100	110	120
11 :	11	22	33	44	55	66	77	88	99	110	121	132
12 :	12	24	36	48	60	72	84	96	108	120	132	144

EXERCICE 3: Nombres premiers

Écrire un programme qui demande un nombre à l'utilisateur et qui teste si ce nombre est premier ou pas.

EXERCICE 4: Nombres amis

Soit n et m , deux entiers positifs. n et m sont dits *amis* si la somme de tous les diviseurs de n (sauf n lui-même) est égale à m et si la somme de tous les diviseurs de m (sauf m lui-même) est égale à n .

Écrire une fonction qui demande à l'utilisateur deux entiers n et m et qui affiche si n et m sont ou non des nombres amis.

Écrire une fonction qui demande à l'utilisateur un entier positif $nmax$ et qui affiche tous les couples de nombres amis (n, m) tels que $n \leq m \leq nmax$.

EXERCICE 5: Racines réelles d'un polynôme du second degré

Ecrivez un programme qui calcule les racines réelles d'un polynôme du second degré. Vous afficherez la valeur exacte des racines, ainsi que leurs valeurs approchées.

EXERCICE 6: Sommes ...

Ecrivez trois versions d'un programme qui calcule la série $S(n) = \sum_{i=1}^n \frac{1}{i}$. La première version utilisera une boucle *for*, la deuxième une boucle *while*, et la troisième une boucle *do while*. Ecrivez également une version qui fait la somme de n à 1 : $S(n) = \sum_{i=n}^1 \frac{1}{i}$. Comparez les résultats des deux méthodes !

EXERCICE 7: Calcul de suite

Calculer les valeurs successives de la suite :

$$u_n = \sqrt{1 + \sqrt{2 + \sqrt{\dots + \sqrt{n}}}}, \text{ pour } 1 \leq n \leq N.$$

EXERCICE 8: Les suites de Syracuse

On se propose de construire un petit programme qui permet d'étudier les suites dites de Syracuse :

$$u_{n+1} = \begin{cases} \frac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

La conjecture de Syracuse dit que quelle que soit la valeur de départ, la suite finit par boucler sur les valeurs 4,2,1,4,2,1,...

1. Construire un programme qui demande une valeur de départ u_0 et affiche les valeurs successives jusqu'à tomber sur la valeur 1 ;
2. modifier le programme pour qu'il compte les itérations, sans affichage intermédiaire ;
3. modifier le programme pour qu'il redemande éventuellement une nouvelle valeur de départ, à l'aide d'une boucle `do ... while`.