

---

# L1 MIASHS - ALGORITHMIQUE

## FICHE EXERCICES 6

### LES BOUCLES

---

#### EXO 1

---

Créez un algorithme qui affiche les entiers de 1 à 100

#### EXO 2

---

Créez un algorithme qui affiche les entiers pairs de 1 à 100

#### EXO 3

---

Créez un algorithme qui affiche les entiers pairs entre 1 et 100, en partant de 100 (100, 98, 96...)

#### EXO 4

---

Créez un algorithme qui affiche les entiers entre 1 et 100 divisibles par 2 et 3

#### EXO 5

---

Dans l'algorithme ci-dessous, réécrivez la boucle POUR sous la format d'un TANTQUE :

```
ALGO boucle
VARIABLES
  i : NUMERIQUE
  j : NUMERIQUE
  s : NUMERIQUE
DEBUT
  s <- 0
  j <- 10
  POUR i ALLANT DE 1 A 10 FAIRE
    s <- s + i*j
    j <- j-1
  FINPOUR
FIN
```

## EXO 6

---

Ecrivez un algorithme qui calcule la somme de  $n$  entiers donnés par l'utilisateur. Deux versions sont demandées :

- La valeur  $n$  est demandée à l'utilisateur via le clavier
- $n$  est donné en argument via les ENTREES.

## EXO 7

---

Ecrire un algorithme qui affiche tous les résultats d'un produit  $a*b$ , où  $a$  et  $b$  varient entre 1 et 12 :

- $1 * 1 = 1$
- $1 * 2 = 2$
- ...
- $6 * 7 = 42$
- $6 * 8 = 48$
- ...
- $12 * 11 = 132$
- $12 * 12 = 144$

## EXO 8

---

Jeu du plus petit/plus grand. Programmez un jeu dans lequel l'utilisateur doit deviner un nombre imaginé par l'ordinateur.

- Pour tirer un nombre au hasard entre 1 et 1000, voici l'instruction à utiliser
  - `nombre <- hasard(1,1000)`
  - `hasard(1,1000)` tire au hasard un nombre entre 1 et 1000, renvoie la valeur tirée, et elle est stockée dans `nombre`.
- Puis l'utilisateur propose une valeur au clavier ; l'ordinateur lui indique si la valeur à chercher est plus grande ou plus petite ; et l'utilisateur peut choisir une nouvelle valeur, et ainsi de suite
- Quand l'utilisateur a trouvé la bonne valeur, le programme lui indique sa victoire et termine
- Une fois cet algorithme écrit et testé, améliorez-le :
  - L'algorithme doit compter le nombre d'essais et l'indiquer au joueur à la fin
  - L'algorithme doit demander si le joueur veut faire une nouvelle partie. Si oui, on recommence depuis le début, si non le programme termine.