

Exercice I : Quels résultats fournira l'exécution de ce programme

```
Tableau Nombre [5] : numérique
Variable i : numérique
Répéter pour i=1 à 5
    Nombre [i] ← i * i
Fin Pour
Répéter pour i=1 à 5
    Ecrire Nombre [i]
Fin Pour
```

La première boucle avec compteur écrit permet de remplir le tableau nombre avec les valeurs suivantes :

Nombre :

1	4	9	16	25
---	---	---	----	----

La seconde boucle avec compteur écrit les valeurs stockées dans le tableau, d'où les résultats :

1
4
9
16
25

Exercice II : Quels résultats fournira l'exécution de ce programme

```
Variable i : numérique
Tableau c[6] : numérique
Répéter pour i=1 à 6
    Lire c[i]
Fin Pour
Répéter pour i=1 à 6
    c[i] ← c[i] * c[i]
Fin Pour
Répéter pour i=1 à 6
    Ecrire c[i]
Fin Pour
```

Si on fournit ces données

2
5
3
10
4
2

La première boucle avec compteur permet lire les données et de les ranger dans le tableau c :

2	5	3	10	4	2
---	---	---	----	---	---

La seconde boucle avec compteur remplace chaque élément de c par son carré :

4	25	9	100	16	4
---	----	---	-----	----	---

Enfin, la dernière boucle avec compteur écrit les valeurs de c, d'où les résultats :

4
25
9
100
16

Exercice III : Que fournira l'exécution de ce programme

```

Tableau Nombre[6] : numérique
Variable k : numérique
Nombre [1] ← 1
Répéter pour k = 2 à 6
    Nombre [k] ← Nombre [k-1] + 2
Fin Pour
Répéter pour k = 1 à 6
    Ecrire Nombre [k]
Fin Pour

```

Le premier élément du tableau « nombre » se voit affecter la valeur 1. La première boucle avec compteur permet d'affecter des valeurs aux éléments suivants. Chacun d'entre eux a une valeur égale à celle de l'élément qui le précède, augmentée de 2. Voici les résultats obtenus :

```

1
3
5
7
9
11

```

Exercice IV : Que fournira l'exécution de ce programme

```

Tableau Suite [8] : numérique
Variable i : numérique
Suite [1] ← 1
Suite [2] ← 1
Répéter pour i = 3 à 8
    Suite [i] ← Suite [i-1] + Suite [i - 2]
Fin Pour
Répéter pour i = 1 à 8
    Ecrire Suite [i]
Fin Pour

```

Les deux premiers éléments du tableau « Suite » se voient affecter la valeur 1. La première boucle avec compteur permet d'affecter des valeurs aux éléments suivants. Chacun d'entre eux a une valeur égale à la somme des valeurs des deux éléments qui le précèdent. Voici les résultats obtenus :

```

1
1
2
3
5
8
13
21

```

Exercice V : Soit t un tableau de 200 éléments réservé par :

Tableau t[200] : numérique

Ecrire les instructions permettant d'obtenir la somme (Spos) des éléments positifs et la somme (Sneg) des éléments négatifs de ce tableau t.

```

Variables Spos, Sneg, i : numériques
Tableau t[20] : numérique
Spos ← 0
Sneg ← 0
Répéter pour i=1 à 200

```

```

    Si t[i] > 0 alors
        Spos ← Spos + t[i]
    Sinon
        Sneg ← Sneg + t[i]
    Fsi
Fpour
Ecrire « Somme des positifs », Spos
Ecrire « Somme des négatifs », Sneg

```

Exercice VI : Ecrire les instructions permettant de déterminer simultanément la plus grande valeur (Max) et la plus petite (Min) d'un tableau t contenant 20 valeurs.

```

Variables Max, Min, i : numériques
Tableau t[20] : numérique
...
Max ← t[1]
Min ← t[1]
i ← 1
Répéter
    i ← i + 1
    Si t[i] > Max alors Max ← t[i] Fsi
    Si t[i] < Min alors Min ← t[i] Fsi
Jusqu'à i=20

```

Exercice VII : Ecrire les instructions permettant de déterminer la position (c'est à dire la valeur de l'indice) du plus grand élément d'un tableau t contenant 20 valeurs.

Nous procédons par itération en supposant que le problème est résolu pour les k premiers éléments de t. Autrement dit, nous choisissons comme état intermédiaire :

i	Max	Pos
k	Plus grande valeur des k premiers éléments de t	Indice correspondant à cette valeur

Le passage à l'état suivant :

i	Max	Pos
k+1	Plus grande valeur des k+1 premiers éléments de t	Indice correspondant à cette valeur

Se fait par les instructions :

<pre> i ← i + 1 Si t[i] > Max alors Max ← t[i] Pos ← i Fsi </pre>
--

L'arrêt et l'initialisation d'une telle itération vous sont maintenant familières. D'où le programme :

```

Variables i, Max, Pos : numériques
Tableau t[20] : numérique
...
i ← 1
Max ← t[i]
Pos ← 1
Répéter
    i ← i + 1
    Si t[i] > Max alors
        Max ← t[i]
        Pos ← i
Fsi

```

Jusqu'à $i = 20$

Exercice VIII : Quels résultats fournira ce programme :

```
Variables i, j : numériques
Ecrire « ** Début ** »
Répéter pour i = 1 à 2
  Ecrire « i avant », i
  Répéter pour j = 1 à 3
    Ecrire « ...i = », i, « j = », j
  Fin Pour
  Ecrire « i après », i
Fin Pour
Ecrire « ** Fin ** »
```

**** début ****

i avant 1

... i = 1 j = 1

... i = 1 j = 2

... i = 1 j = 3

i après 1

i avant 2

... i = 2 j = 1

... i = 2 j = 2

... i = 2 j = 3

i après 2

**** Fin ****

Exercice IX : On dispose des notes de 25 élèves ; chaque élève peut avoir une ou plusieurs notes (mais toujours au moins une). Ecrire un programme permettant d'obtenir la moyenne de chaque élève. On veut que données et résultats se présentent ainsi :

Notes de l'élève numéro 1

12

10

-1

Moyenne : 11

Notes de l'élève numéro 2

...

Les parties en italiques correspondent aux données tapées par l'utilisateur. La valeur -1 sert de critère de fin de notes pour chaque élève.

Première ébauche du programme :

Répéter pour $i = 1$ à 25

 Déterminer la moyenne de l'élève numéro i

 Pour chaque élève, une boucle jusqu'à permet de lire les différentes notes, de les compter (dans n) et d'en faire la somme (dans Som). Nous ne devons pas oublier que la note fictive -1 a aussi été prise en compte.

D'où le programme :

```
Variables i, n, Som, Note, Moyenne : numériques
Répéter pour i=1 à 25
  Ecrire « Notes de l'élève numéro », i
```

```

Som ← 0
n ← 0
Répéter
    Lire Note
    Som ← Som + Note
    n ← n + 1
jusqu'à Note = -1
n ← n - 1
Som ← Som + 1
Moyenne ← Som / n
Ecrire « Moyenne », Moyenne
FinPour

```

Rem : Ce programme n'est pas protégé contre les erreurs telles que :

- Aucune note (valeur -1 fournie immédiatement après la question : note de l'élève...).
- Dans ce cas, le calcul de la moyenne conduit à une division par zéro.
- Valeurs négatives autres que -1. Elles sont comptabilisées comme les autres notes. On peut éviter ce problème en choisissant comme critère d'arrêt de la boucle jusqu'à :

Note < 0

Il faut alors ne pas oublier de remplacer Som ← Som + 1 par Som ← Som - Note.

Exercice X : On souhaite faire la même chose que dans l'exercice précédent mais, cette fois, on veut que données et résultats se présentent ainsi :

```

Notes de l'élève numéro 1
12
10
-1
Notes de l'élève numéro 2
...
Notes de l'élève numéro 25
15
-1
Moyennes
Elève numéro 1 : 11
...
Elève numéro 25 : 15
Moyenne de la classe : 12, 3

```

Autrement dit, il est nécessaire de conserver en mémoire les moyennes de chaque élève avant de les écrire...

Il est nécessaire d'utiliser un tableau Tmoy pour conserver les 25 moyennes, après calcul.

Voici une première ébauche du programme. La variable Som sert à calculer la somme des moyennes. La moyenne de la classe s'obtient en divisant par 25 la valeur de Som.

```

Som ← 0
Répéter pour i = 1 à 25
    déterminer la moyenne de l'élève i
    Tmoy[i] ← Moy
    Som ← Som + Moy
FinPour
Ecrire « Moyenne »
Répéter pour i=1 à 25
    Ecrire la moyenne de l'élève i
FinPour
Calculer et écrire la moyenne de la classe.

```

```

Variables i, n, Som, SomElev, Note, Moyenne, MoyenneClasse :
numériques
Tableau Tmoy[25] : numériques
Som ← 0
Répéter pour i=1 à 25
    Ecrire « Notes de l'élève numéro »,i
    Som ← 0
    n ← 0
    Répéter
        Lire Note
        SomElev ← SomElev + Note
        n ← n +1
    jusqu'à Note = -1
    n ← n - 1
    SomElev ← SomElev + 1
    Moyenne ← SomElev / n
    Tmoy[i] ← Moyenne
    Som ← Som + Moyenne
FinPour
Ecrire « Moyenne »
Répéter pour i = 1 à 25
    Ecrire « Elève numéro i », i, Moy[i]
FinPour
MoyenneClasse ← Som / 25
Ecrire « Moyenne de la classe », MoyenneClasse

```

Exercice XI : Déterminer tous les nombres premiers inférieurs à 10000 par la méthode du crible d'Erathostène qui consiste à « barrer » de la liste des nombres, tous les multiples de 2, puis les multiples de 3, puis tous les multiples de 5 (on « saute » 4 puisqu'il est déjà barré), etc. Si l'on part d'un tableau de booléens, tous initialisés à Vrai, l'opération « barrer » un nombre consiste à lui affecter la valeur Faux.

```

Variables i, j, N : entiers
Tableau P [10000] : booléens
// initialisation du tableau à vrai
// On n'utilise que les valeurs de 1 à  $\sqrt{10000} = 100$ 
N ← 100
répéter pour i = 2 à 100
    P[i] ← Vrai
FinPour

// on barre maintenant les multiples de i
i ← 2
Tant que i < N faire
    Si P[i]=Vrai alors
        j ← i * i // on barre seulement à partir de  $i^2$ 
        Tant Que (j <= 10000) faire
            P[j] ← Faux
            j ← j + i
        FinTantQue
    Fsi
    i ← i + 1
FinTantQue

// Affichage des nombres premiers
Répéter pour i = 2 à 10000
    Si P[i] = Vrai alors Ecrire (i)

```

FinPour
Fsi