Pages 21 à 23

EXERCICE 39 Exécuter et comprendre un programme

La fonction fac, dont l'argument a est un entier naturel non nul, est programmée ci-dessous en langage Python.

```
def fac(a):
    b=1
    for i in range(1,a+1):
        b=b*i
    return(b)
```

1. a. Que renvoie fac(3) ?

Puisque $a+1=4$, l'entier i varie de 1 à 3. Le nombre b vaut 1 au départ, puis $1\times1=1$, puis $1\times2=2$, puis $2\times3=6$. Ainsi, fac(3) renvoie 6.

b. Que renvoie fac(6) ?

Après la valeur initiale 1, b prend successivement les valeurs 1, 2, 6, 24, 120 et 720. Donc fac(6) renvoie 720.

2. Comment peut-on calculer le produit $1\times2\times3\times4\times5$ en utilisant la fonction fac ?

On saisit fac(5) dans la console.

3. Que calcule cette fonction pour un entier a non nul ?

Pour un entier a non nul, cette fonction calcule le produit des entiers 1, 2, 3... jusqu'à a.

EXERCICE 40 Compléter et programmer un algorithme

En prévision d'une course de vélo, Fanny suit le programme d'entraînement suivant sur douze samedis : elle parcourt 25 kilomètres le premier samedi, puis augmente chaque semaine de 11 kilomètres la distance parcourue.

1. Déterminer la distance D parcourue le deuxième samedi et la distance totale T parcourue au bout de deux samedis d'entraînement.

$D=25+11=36$. Le deuxième samedi, Fanny parcourt 36 kilomètres.

Le premier samedi, Fanny parcourt 25 kilomètres, le deuxième samedi elle parcourt 36 kilomètres, donc la distance totale parcourue au bout de deux samedis est $T=25+36$, soit 61 kilomètres.

2. Compléter l'algorithme ci-dessous afin que la variable T contienne en fin d'algorithme la distance totale parcourue au bout des douze samedis d'entraînement.

$D \leftarrow$ 25
$T \leftarrow$ 25
Pour I variant de 2 à 12
  $D \leftarrow D+11$
  $T \leftarrow T+D$
Fin Pour

3. Programmer une fonction sans argument en langage Python qui retourne la distance totale parcourue à la fin des douze samedis d'entraînement.

```
def distance():
    D=25
    T=25
    for i in range(2,13):
        D=D+11
        T=T+D
    return(T)
```

La distance totale parcourue à la fin des douze samedis d'entraînement est 1 026 kilomètres.

EXERCICE 41

Analyser une situation et compléter un programme

L'an prochain, Oscar consacrera 1 800 € à ses loisirs, puis pour faire des économies, il prévoit à partir de l'année suivante de réduire chaque année de 5 % les dépenses de ce secteur.

1. Calculer le budget loisirs B d'Oscar dans deux ans.

Le budget loisirs B d'Oscar dans deux ans sera de 1 710 €.

2. Quel budget total aura-t-il consacré à ses loisirs au cours des deux prochaines années ?

Ce budget total sera de : 1 800 + 1 710 = 3 510 €.

3. Compléter le programme ci-dessous écrit en langage Python afin que la fonction **budget** retourne le budget total T qu'Oscar aura consacré à ses loisirs au cours des dix prochaines années s'il respecte son plan d'économies.

```
def budget():
    B=1800
    T=0
    for i in range (1,11):
        T=T+B
        B=0.95*B
    return(T)
```

EXERCICE 42

Comprendre et programmer un algorithme

Soit n un entier naturel non nul. On pose $S = 1 + 2 + \ldots + n$.

1. Calculer les sommes $1 + 2$; $1 + 2 + 3$ et $1 + 2 + 3 + 4$, puis calculer S lorsque $n = 5$.

$1 + 2 = 3$; $1 + 2 + 3 = 6$; $1 + 2 + 3 + 4 = 10$; $1 + 2 + 3 + 4 + 5 = 15$.

2. On considère l'algorithme ci-dessous.

$S \longleftarrow 0$
Pour i variant de 1 à n
  $S \longleftarrow S + i$
Fin Pour

Exprimer en fonction de n, la valeur que contient la variable S en fin d'algorithme.

La valeur que contient la variable S est celle de la somme $1 + 2 + 3 + 4 + \ldots + n$.

3. Programmer une fonction d'argument n qui retourne la valeur de la variable S de l'algorithme précédent. En déduire la somme des 1 000 premiers entiers naturels non nuls.

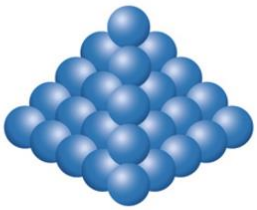
```
def somme(n):
    S=0
    for i in range(1,n+1):
        S=S+i
    return(S)
```

La somme des 1 000 premiers entiers non nuls est :

$1 + 2 + 3 + \ldots + 1\,000 = 500\,500$.

EXERCICE 43

Analyser une situation et écrire un programme



On empile des sphères, formant ainsi une pyramide de base carrée.

1. Déterminer le nombre de sphères nécessaires pour constituer une pyramide de deux « niveaux ».

Il faut cinq sphères : quatre au 1er niveau et une au 2e niveau.

2. Déterminer le nombre de sphères nécessaires pour constituer une pyramide de cinq « niveaux ».

Il faut 55 sphères : une au 5e niveau, 4 au 4e, 9 au 3e, 16 au 2e et 25 au 1er.

3. Sachant qu'il faut 140 sphères pour construire une pyramide de sept « niveaux », déterminer le nombre de sphères nécessaires pour construire une pyramide de huit « niveaux ».

$140 + 8^2 = 204$, donc il faut 204 sphères.

4. En utilisant la structure ci-dessous, compléter le programme de la fonction nb_sphères d'argument un entier naturel non nul N et qui retourne le nombre B de sphères nécessaires pour constituer une pyramide comportant N « niveaux ».

```
def nb_sphères(N):
    B=0
    for i in range (1,N+1):
        B=B+i**2
    return(B)
```

EXERCICE 44

Diviseur d'un entier

Comprendre et modifier un programme

1. À l'aide d'une calculatrice, on a programmé en langage Python la fonction nbdv dont l'argument est un entier naturel a non nul. (Rappel : a%i donne le reste de la division de a par i.)

```
Nbdv.py        001/007
def nbdv(a):
  n=0
  for i in range(1,a+1):
      if a%i==0:
        n=n+1
  return(n)
```

FILE RUN SYMBOL CHAR A⇔a ▷

a. Que signifie l'instruction a%i==0 ? a%i==0 teste si l'entier a est divisible par i.

b. Que renvoie nbdv(6) ?

Puisque $a + 1 = 7$, l'entier i varie de 1 à 6. Pour $i = 1$, le test a%i==0 est vérifié, donc n prend la valeur 1.

Puis le test est vérifié quand i vaut 2, 3 et 6, donc nbdv(6) renvoie 4.

c. Que renvoie cette fonction pour un entier naturel a non nul ?

Cette fonction renvoie le nombre de diviseurs (positifs) de a.

2. Écrire le programme d'une fonction nbpremier d'argument un entier naturel n non nul qui retourne le nombre de nombres premiers inférieurs ou égaux à n. On pourra utiliser la fonction nbdv de la question **1.** On rappelle qu'un entier naturel a est dit premier si et seulement s'il a exactement deux diviseurs positifs.

```
def nbpremier(n):
    c=0
    for a in range(1,n+1):
        if nbdv(a)==2:
            c=c+1
    return(c)
```