

Algorithmique 2 : programmes élémentaires

Exercice 1 : somme des n premiers entiers

Ecrire une suite d'instructions qui saisit un entier $n \geq 1$ et qui calcule la somme $1 + 2 + \dots + n$.

Exercice 2 : carrés successifs

Ecrire une suite d'instructions qui saisit deux nombres entiers n et p , puis qui fournit les « carrés successifs » de n jusqu'au p -ième carré obtenu, c'est-à-dire les valeurs de $n^{(2^k)}$ pour $k = 1, 2, \dots, p$.

Exercice 3 : diviseurs d'un entier

Ecrire une suite d'instructions qui saisit un nombre entier n et fournit tous ses diviseurs (on pourra utiliser l'instruction $n\%p$ qui donne le reste (entier) de la division de n par p).

Exercice 4 : deviner un nombre

En Python, l'instruction « `randint(0,100)` » renvoie un entier aléatoire entre 0 et 100 et elle s'utilise comme dans cet exemple :

```
>>> from random import *
>>> randint(0,100)
92
>>> randint(0,100)
30
```

En utilisant cette fonction, écrire une suite d'instructions qui choisit au hasard un entier de 0 à 100 et qui permet à l'utilisateur de le trouver en lui indiquant à chaque essai si le nombre proposé est trop grand ou trop petit.

Exercice 5 : simulation d'un lancer d'un dé

En Python, la fonction « `random` » renvoie un nombre réel aléatoire dans l'intervalle $[0; 1[$ et elle s'utilise comme dans cet exemple :

```
>>> from random import *
>>> random()
0.9811896401527802
>>> random()
0.9889103424621456
```

En utilisant cette fonction, écrire une instruction qui permet de simuler le lancer d'un dé cubique, c'est-à-dire renvoyer de façon équiprobable un entier de 1 à 6.

Exercice 6 : factorielle d'un entier

Ecrire une suite d'instructions qui calcule la factorielle $n!$ d'un entier $n \geq 1$ entré par l'utilisateur (on rappelle que $n! = 1 \times 2 \times 3 \times 4 \times \dots \times (n-1) \times n$ et que $0! = 1$).

Exercice 7 :

Que fait le programme suivant ?

```
a=1
b=2
if a>= b :
    a=b
else :
    b=a
```

Exercice 8 :

Quelle est la valeur de y à la fin du programme suivant ?

```
x=-3
if x<-4 : y=0
elif x<-3 : y=4-x
elif x<-1 : y=x*x+6*x+8
elif x<3 : y=2-x
else : y=-2
```

Exercice 9 :

Quelle est la valeur de y à la fin du programme suivant ?

```
x=3
y=-2
if x<y : y=y-x
elif x==y : y=0
else : y=x-y
```

Exercice 10 : valeur absolue

Ecrire une série d'instructions affichant la valeur absolue d'un nombre x.

Exercice 11 : admission à un examen

Ecrire une série d'instructions qui saisit une note a (sur 20), puis qui affiche « ajourné » si $a < 8$, « oral » si $8 \leq a < 10$ et « admis » si $a \geq 10$.

Exercice 12 : ouverture d'un guichet

Ecrire une série d'instructions qui détermine si un guichet est ouvert ou fermé, selon les jours de la semaine et les heures de la journée (de 0 à 24 h), sachant que le guichet est ouvert tous les jours de 8 h à 13 h et de 14 h à 17 h, sauf le samedi après-midi et toute la journée du dimanche.

Exercice 13 : assurance

Une compagnie d'assurances effectue des remboursements en laissant une somme, appelée franchise, qui reste à la charge de l'assuré. Cette franchise représente 10% du montant des dommages sans toutefois pouvoir être inférieure à 15 € et supérieure à 500 €. Ecrire une série d'instructions qui demandent à l'utilisateur de saisir le montant des dommages et lui affichent le montant remboursé ainsi que le montant de la franchise.

Exercice 14 : pièces de monnaie

On veut réaliser une somme d'argent (limitée à 10 €) avec des pièces de 0,01, 0,02, 0,05, 0,10, 0,20, 0,50, 1 et 2 €, en utilisant le moins de pièces possible. Par exemple, $S = 7,29$ € se paie avec trois pièces de 2, une pièce de 1, une pièce de 0,20, une pièce de 0,05 et deux pièces de 0,02.

Ecrire un algorithme demandant à l'utilisateur de saisir une somme d'argent, puis détermine le détail des pièces à utiliser pour constituer la somme saisie en utilisant le moins de pièces possible.