Exercice 1 Lancer 100 fois un dé et compter le nombre de 6

script disponible BP_ exoresolu1.py

Énoncé

Écrire un algorithme pour simuler 100 lancers d'un dé cubique, bien équilibré, et compter le nombre de 6 obtenus sur les 100 lancers. L'implémenter en Python.

Solution

Analysons l'énoncé. Il s'agit de simuler100 fois un lancer de dé. On connaît donc le nombre de répétitions et nous allons utiliser une boucle bornée « Pour ... »

Nous devons également compter le nombre de fois où l'on obtient un 6, donc créer une variable pour cela et tester à chaque lancer si on obtient 6. Si c'est le cas, il faut augmenter de 1 le nombre de 6 obtenus.

Algorithme	Programme en Python
VARIABLES : *d* entier entre 1 et 6 , *nb6* entier INITIALISATION : *nb6* prend la valeur 0 TRAITEMENT : Pour *k* de 1 à 100 Faire : Affecter à *d* un entier aléatoire entre 1 et 6 Si *d* = 6 Alors *nb6* prend la valeur *nb*6 + 1 Fin Pour SORTIE : Afficher *nb6*	

```
import random
#  nécessaire pour disposer de randint()

nb6 = 0
for k in range(1, 101):
    d = random.randint(1, 6)
    if d == 6:
        nb6 = nb6 + 1
print(nb6)
```

Remarque : le nom d'une variable doit commencer par une lettre, même si elle peut contenir des chiffres ensuite.

Exercice 2 Calculer une somme

Énoncé

La légende du jeu d'échec dit que son inventeur, demanda en remerciement que l'on dépose 1 grain de blé sur la 1re case de l'échiquier, 2 grains sur la 2e case, 4 grains sur la 3e, 8 grains sur la 4e, etc. en doublant le nombre de grains à chaque fois. Écrire un programme qui calcule le nombre total de grains (un échiquier comporte 64 cases).

Solution

Analysons l'énoncé. Il faut répéter une action pour chacune des 64 cases de l'échiquier, nous allons donc utiliser une boucle bornée.

On doit savoir, à chaque case, le nombre de grains à déposer et les ajouter au nombre total de grains déjà déposés.

On va donc créer :

– une variable nb contenant, au fur et à mesure, le nombre de grains à poser sur chaque case.

Au début, pour la 1re case, nb prend la valeur 1, puis la valeur de nb est multipliée par 2 à chaque case.

– une variable S contenant, au fur et à mesure, le nombre total de grains déjà déposés. Au début, pour la 1re case, *S* prend la valeur 1. A chaque case, il faudra augmenter la valeur de S du nombre de grains déposés dans la case.

D'où le programme ci-contre :

Remarque : range(1, 64) crée la liste des entiers de 1 à 63, il y a donc bien 63 répétitions.

```
# A la première étape, sur la première case
nb, S = 1, 1
# On ajoute pas à pas pour les 63 cases suivantes
for k in range(1,64):
    nb = nb * 2
    S = S + nb
print(S)
```

On obtient 18 446 744 073 709 551 615 grains !