

Types et valeurs de base – Exercices – Devoirs

Exercice 1 corrigé disponible

Dans cet exercice, il vous est demandé de programmer une version simplifiée du black jack (jeu de casino).

Règles : Deux joueurs jouent l'un contre l'autre. L'un des joueurs est humain, l'autre est l'ordinateur. Le but du jeu est de marquer plus de points que l'autre, sans dépasser 21, qui est la plus grande valeur possible. Les cartes de 2 à 9 ont leur valeur (donc de 2 à 9 points...). Le 10, le valet, la dame et le roi valent 10 points. L'as (le 1) vaut 11 points. Toutes les cartes sont tirées face visible.

- Les joueurs tirent d'abord une première carte, face visible.
 - Ensuite, le joueur humain tire autant de cartes qu'il le veut. S'il dépasse 21, il a perdu.
 - Le deuxième joueur -l'ordinateur- tire ensuite autant de cartes qu'il le veut. S'il dépasse 21, il a perdu.
 - Si un des joueurs a plus de points que l'autre, il a gagné. Si les deux joueurs ont le même nombre de points, il y a égalité.
1. Écrire une fonction qui tire un nombre aléatoire entre 1 et 13, et renvoie les points correspondant à la valeur de la carte. On considérera que 11, 12 et 13 correspondent respectivement au valet, à la dame et au roi.
 2. Écrire une fonction qui initialise le jeu en tirant une carte pour le joueur humain et une carte pour l'ordinateur. Cette fonction renverra les points correspondant à chaque carte.
 3. Écrire une fonction qui fait jouer le joueur humain. Cette fonction renvoie le nombre de points du joueur humain.
 4. Écrire une fonction qui fait jouer l'ordinateur. Vous programmerez une intelligence artificielle (rudimentaire ☺) qui décide de la stratégie de l'ordinateur. Vous expliquerez cette stratégie dans les spécifications de la fonction. Cette fonction renvoie le nombre de points de l'ordinateur.
 5. Écrire le programme principal, qui organise la partie et détermine le vainqueur.

Exercice 2 corrigé disponible

Écrire une fonction `nombre_de_a(chaine)` qui demande une chaîne de caractères à l'utilisateur, et compte le nombre de caractères « a » ou « A » dans la chaîne. On rappelle qu'une chaîne est un type particulier de liste, que `len(chaine)` donne la longueur de la chaîne et que `chaine[i]` permet d'accéder à l'élément d'indice i de la chaîne. Si vous le souhaitez, plutôt que d'écrire une chaîne vous pouvez utiliser une liste (inconvenient : c'est plus long à taper).

Exemples :

- `nombre_de_a("bonjour Ernest")` renvoie 0
- `nombre_de_a("Salut Bernard")` renvoie 2
- `nombre_de_a("Au revoir Anatole ")` renvoie 3

Exercice 3 corrigé disponible

On dispose d'un dé équilibré à 6 faces et de deux urnes : l'urne U_1 contient deux boules vertes et 3 rouges, et l'urne U_2 contient 1 boule verte et deux rouges. On lance le dé et si le résultat est 1 ou 2 alors on tire une boule dans l'urne U_1 , sinon on tire dans l'urne U_2 . On considère que la partie est gagnante si on tire une boule verte.

1. Écrire un algorithme en Python permettant de simuler cette partie.
2. Modifier cet algorithme pour qu'il simule n parties et compte le nombre de parties gagnantes.

Exercice 4 corrigé disponible

Cet exercice est un QCM où chaque question comporte 4 ou 5 réponses. Une seule est bonne. Indiquer la réponse correcte

1. On donne un programme faux, qui est censé donner les valeurs de $f(x)=x^2+3x+1$ pour x de 1 à 20.

Quelle version fonctionne correctement ?

Code A	Code B
<pre>def parabole(x): y =x**2+3*x + 1 return x = 1 for i in range(21): x = x + i print(parabole(x))</pre>	<pre>def parabole(x): y =x**2 + 3*x + 1 return (x,y) x = 1 for i in range(20): x = i print(parabole(x))</pre>
Code C	Code D
<pre>def parabole(x): y =x**2 + 3*x + 1 return (y) x = 1 for i in range(1,21): x = i print(parabole(x))</pre>	<pre>def parabole(x): y =x**2 + 3*x + 1 return y for i in range(20): x = i print(parabole(x))</pre>

2. Le programme suivant affiche :

```
somme = 0.1 + 0.1 + 0.1
if somme == 0.3 :
    print("somme = ",somme)
else :
    print("résultat = ",somme)
```

- A. somme = 0.3 B. somme = 0.1 + 0.1 + 0.1
C. résultat = 0.1 + 0.1 + 0.1
D. résultat = 0.300000000000000004 E. un message d'erreur

3. Que donne l'exécution du programme suivant ?

```
x = 5
y = 4
if x == y - 1 :
    y = x
else :
    x = y
if y == x - 1 :
    x = x + 1
else :
    y = y + 1
```

- A. $x=5$ et $y=5$ B. $x=4$ et $y=4$ C. $x=5$ et $y=4$ D. $x=4$ et $y=5$
E. autre réponse

4. Qu'affiche le programme suivant

```
somme = 0
for i in range(5):
    somme = i
print(somme, end = "")
```

- A. 5 B. 4 C. 10 D. 0 1 2 3 4

5. Quel est le type de la variable nombre = "3.0" ?

- A. string B. float
C. integer D. autre réponse

6. Quelle valeur contient la variable x après exécution du programme suivant ?

```
x = 400
while x>=25 :
    x = x//2
```

- A. 25 B. 200 C. 12 D. 13 E. 12,5

Exercice 5 corrigé disponible

On donne une chaîne de caractères, dans laquelle figurent également des chiffres. Écrire une fonction `somme_1_2(chaîne)` qui renvoie la somme des 1 et 2 figurant dans la chaîne. On rappelle qu'une chaîne est un type particulier de liste, que `len(chaîne)` donne la longueur de la chaîne et que `chaîne[i]` permet d'accéder à l'élément d'indice i de la chaîne.

Exemples :

```
somme_1_2("j'ai mangé 3 œufs") renvoie 0
somme_1_2("j'ai mangé 12 œufs, 1 bœuf, 5 gâteaux et un chapeau ")
renvoie 4
```

Exercice 6 corrigé disponible

En 1202, le mathématicien Léonardo Fibonacci inventa l'énigme suivante : Un homme met un couple de lapins dans un lieu isolé de tous les côtés par un mur. Combien de couples obtient-on en un an si chaque couple engendre tous les mois un nouveau couple à compter du troisième mois de son existence ?

1. Si l'on connaît le nombre de couples de lapins au mois n et au mois $n + 1$, comment connaître le nombre de couples de lapins au mois $n + 2$?
2. Résoudre l'énigme.
3. Plus généralement, écrire une fonction `fibonacci` qui calcule le nombre de couples de lapins au bout d'un nombre n de mois donné en argument.

Exercice 7 corrigé disponible

On représente un brin d'ADN par une chaîne de caractères qui peut contenir quatre caractères différents : "A" (Adénine), "C" (Cytosine), "G" (Guanine) et "T" (Thymine).

1. Écrire une fonction `estADN` qui prend en argument une chaîne de caractères et renvoie `True` si cette chaîne correspond à un brin d'ADN et qui renvoie `False` sinon.

Contrat:

Par exemple, `estADN("TTGAC")` et `estADN("GCAATAG")` renvoient `True` mais `estADN("AMOG")` et `estADN("CaTg")` renvoient `False`.

2. Écrire une fonction `masseMolaire` qui calcule la masse molaire d'une séquence ADN passée en argument. Chaque lettre a une masse donnée : "A" (135 g/mol); "T" (126 g/mol); "G" (151 g/mol); "C" (111 g/mol). La masse totale est la somme des masses des lettres de la séquence.

Contrat:

Par exemple, `masseMolaire("AGATC")` renvoie (135 + 151 + 135 + 126 + 111) g/mol, c'est-à-dire 658 g/mol.

3. Chaque base possède une base complémentaire avec laquelle elle peut s'associer : "A" et "T" sont complémentaires, et "C" et "G" sont complémentaires. Écrire une fonction `brinComp` qui étant donné un brin d'ADN `b` calcule et renvoie son brin complémentaire, c'est à dire le brin constitué des bases complémentaires de `b`.

Contrat:

Par exemple, `brinComp("A")` renvoie "T" et `brinComp("AAGT")` renvoie "TTCA".

4. (Bonus, ***) Écrire une fonction `sous_séquence` qui prend en argument deux chaînes de caractères représentant des brins d'ADN et renvoie `True` si le premier brin est une sous-séquence du deuxième, et qui renvoie `False` sinon.

Contrat:

Par exemple, `sous_séquence("ATC", "GGTATCG")` renvoie `True` et `sous_séquence("GC", "AAT")` renvoie `False`.

Exercice 8 corrigé disponible

Donner les lignes d'une fonction Python, la plus simple possible permettant de d'apporter une solution aux problèmes mathématiques suivants :

1. retourner le PGCD et le PPCM de deux nombres entiers A et B
2. Quelle est une valeur approchée de π en utilisant la méthode Monte-Carlo ?
3. Quelle est une valeur approchée de \sqrt{x} en utilisant la méthode de Héron ?
4. Retourner la solution de l'équation $x^3 = k$ en utilisant la méthode de dichotomie

Exercice 9 corrigé disponible

Question A.1

Combien d'entiers positifs ou nuls (entiers non signés) peut-on représenter en machine sur 32 bits ?

Réponses

A $2^{32} - 1$

B 2^{32}

C 2×32

D 32^2

Question A.2

Quel est un avantage du codage UTF8 par rapport au codage ASCII ?

Réponses

A il permet de coder un caractère sur un octet au lieu de deux

B il permet de coder les majuscules

C il permet de coder tous les caractères

D il permet de coder différentes polices de caractères

Question A.3

Quelle est la représentation en binaire signé en complément à 2 de l'entier -1 sur un octet ?

Réponses

A 1000 0000

B 1000 0001

C 1111 1110

D 1111 1111

Question A.4

Quelle est la plage des valeurs entières (positifs ou négatifs) que l'on peut coder sur un octet (8 bits) en complément à 2 ?

Réponses

A -127 à 128

B -128 à 127

C -255 à 128

D -256 à 127

Question A.5

Dans quel système de numération 3F5 représente-t-il un nombre entier ?

Réponses

A binaire (base 2)

B octal (base 8)

C décimal (base 10)

D hexadécimal (base 16)

Question A.6

Parmi les quatre propositions, quelle est celle qui correspond au résultat de l'addition en écriture binaire $1101\ 1001 + 11\ 0110$?

Réponses

A 1000 1111

B 10 0000 1111

C 1 0000 1111

D 1 1000 0111

Exercice 10 corrigé disponible

Question A.1

Quel est le plus grand entier positif que l'on peut coder sur un mot de 16 bits ?

Réponses

A $2^{15} - 1 = 32767$

B $2^{15} = 32768$

C $2^{16} - 1 = 65535$

D $2^{16} = 65536$

Question A.2

À quelle affectation sont équivalentes les instructions suivantes, où a, b sont des variables entières et c une variable booléenne ?

if a==b:

c = True

elif a > b+10:

c = True

else:

c = False

Réponses

A $c = (a==b) \text{ or } (a > b+10)$

B $c = (a==b) \text{ and } (a > b+10)$

C $c = \text{not}(a==b)$

D $c = \text{not}(a > b+10)$

Question A.3

A et B sont deux propositions vraies. Laquelle des ces propositions est-elle également vraie ?

Réponses

A ((non A) et B) ou (A et (non B))

B ((non A) ou B) et ((non A) ou (non B))

C ((non A) ou B) et (A ou (non B))

D ((non A) et B) et (A et (non B))

Question A.4

Le résultat de la soustraction en binaire $101001 - 101$ est égal au nombre binaire :

Réponses

A 100900

B 101110

C 100100

D 100110

Question A.5

Quelle est la représentation en binaire de l'entier 64 sur un octet ?

Réponses

A 0101 0000

B 1100 0100

C 0100 0000

D 0000 1100

Question A.6

Quel est l'entier codé sur 4 bits en complément à 2 par 1101 ?

Réponses

A -6

B -3

C 13

D 14

Exercice 11 corrigé disponible

Quelle est l'écriture décimale de l'entier dont la représentation en binaire non signé est 0001 0101 ?

Réponses

A 15

B 21

C 111

D 420

Question A.2

Dans le bloc <head> d'un fichier HTML, afin d'encoder les caractères avec le standard Unicode/UTF-8 on insère la ligne :

```
<meta http-equiv="Content -Type" content="text/html; charset=UTF-8">
```

Pourquoi cela ?

Réponses

A UTF-8 est l'encodage Linux

B ASCII est une vieille norme, il est temps d'en changer

C UTF-8 est une norme conçue pour permettre un affichage correct des caractères spéciaux sur tout système d'exploitation

D UTF-8 est un encodage qui protège mieux contre le piratage informatique

Question A.3

Un seul des réels suivants (écrits en base 10) n'a pas une écriture finie en base 2. Lequel ?

Réponses

A 1,25

B 1,5

C 1,6

D 1,75

Question A.4

Combien de bits faut-il au minimum pour coder le nombre décimal 4085 ?

Réponses

A 4

B 12

C 2042

D 2043

Question A.5

Soient a et b deux booléens. L'expression booléenne $\text{NOT}(a \text{ AND } b) \text{ OR } a$ est équivalente à :

Réponses

A False

B True

C $\text{NOT}(b)$

D $\text{NOT}(a) \text{ OR } \text{NOT}(b)$

Question A.6

Quelle est la plage des valeurs entières (positifs ou négatifs) que l'on peut coder sur un octet (8 bits) en complément à 2 ?

Réponses

A -127 à 128

B -128 à 127

C -255 à 128

D -256 à 127