Lycée Privé d’Avesnières Jeudi 6 mai 2021

|  |  |
| --- | --- |
| NOM : | QCM 6.1 |
| Prénom : | 55 minutes |
| Classe : Première, spécialité NSI | Calculatrice interdite |

* Ce QCM se compose de 20 questions réparties en 4 questions sur 5 thèmes.
* Pour chaque question, une seule des quatre réponses est exacte.
  + Une réponse correcte rapporte 3 points.
  + Une réponse fausse retire 1 point.
  + L’absence de réponse n’apporte pas de point et n’enlève pas de point.
  + Le total est ramené sur 20 points.

**Thème A : types de base** choisies [dans](http://www.astrovirtuel.fr/nsi/premiere/bns_e3c_nsi.htm) **1.1 Ecriture d’un entier positif - 1.3 notion des nombres flottants et 1.4 Représentation d’un texte en machine**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Question | 1 **Q24 (1.1)** | 2 **Q3 (1.3)** | 3 **Q9 (1.4)** | 4 **Q57 (1.1)** |
| Réponse choisie | **D** | **C** | **B** | **B** |

**Thème B : types construits** choisies [dans](http://www.astrovirtuel.fr/nsi/premiere/bns_e3c_nsi.htm) **2.1 Python : les séquences (tuples et tableaux)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Question | 1 **Q46** | 2 **Q87** | 3 **Q90** | 4 **Q100** |
| Réponse choisie | **C** | **C** | **C** | **B** |

**Thème E : architectures matérielles et systèmes d'exploitation** choisies [dans](http://www.astrovirtuel.fr/nsi/premiere/bns_e3c_nsi.htm) **5.1 Modèle d’architecture de Von Neumann (portes logiques)** et **5.3 Systèmes d’exploitation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Question | 1 **Q16 (5.1)** | 2 **Q14 (5.3)** | 3 **Q22 (5.1)** | 4 **Q26 (5.3)** |
| Réponse choisie | **D** | **D** | **D** | **B** |

**Thème F : langages et programmation** choisies [dans](http://www.astrovirtuel.fr/nsi/premiere/bns_e3c_nsi.htm) **6.1 Python : les bases** et **6.2 Langages**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Question | 1 **Q8 (6.1)** | 2 **Q9 (6.2)** | 3 **Q16 (6.2)** | 4 **Q24 (6.2)** |
| Réponse choisie | **B** | **B** | **A** | **C** |

**Thème G : algorithmique** choisies [dans](http://www.astrovirtuel.fr/nsi/premiere/bns_e3c_nsi.htm) **7. Algorithmique**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Question | 1 **Q46** | 2 **Q45** | 3 **Q26** | 4 **Q27** |
| Réponse choisie | **B** | **B** | **B** | **C** |

**Thème A : types de base**

**Question A.1**

Quelle est, en écriture décimale, la somme d'entiers dont l'écriture en base 16 (hexadécimale) est 2A + 2 ?

***Réponses* :**

A- 22

B- 31

C- 49

D- 44

**Question A.2**

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.3**

Quel est le nombre minimum de bits qui permet de représenter les 26 lettres majuscules de l'alphabet ?

***Réponses* :**

A- 4

B- 5

C- 25

D- 26

**Question A.4**

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

***Réponses* :**

A- 232-1

B- 232

C- 2 x 32

D- 322

**Thème B : types construits**

**Question B.1**

Quelle est la valeur de l'expression [[n, n+2] for n in range(3)] ?

***Réponses* :**

A- [0, 2, 1, 3, 2, 4]

B- [1, 3, 2, 4, 3, 5]

C- [[0,2], [1,3], [2,4]]

D- [[1,3], [2,4], [3,5]]

**Question B.2**

Soit le tableau défini de la manière suivante : tableau = [[1,3,4], [2,7,8], [9,10,6], [12,11,5]]

On souhaite accéder à la valeur 12, on écrit pour cela :

***Réponses* :**

A- tableau[4][1]

B- tableau[1][4]

C- tableau[3][0]

D- tableau[0][3]

**Question B.3**

- Quelle est la valeur de la variable image après exécution du programme Python suivant ?

image = [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]

for i in range(4):

for j in range(4):

if (i+j) == 3:

image[i][j] = 1

Quelle est la valeur de la variable a à la fin de cette exécution ?

***Réponses* :**

A- [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [1, 1, 1, 1]]

B- [[0, 0, 0, 1], [0, 0, 0, 1], [0, 0, 0, 1], [0, 0, 0, 1]]

C- [[0, 0, 0, 1], [0, 0, 1, 0], [0, 1, 0, 0], [1, 0, 0, 0]]

D- [[0, 0, 0, 1], [0, 0, 1, 1], [0, 1, 1, 1], [1, 1, 1, 1]]

**Question B.4**

On exécute le code suivant :

def maxi(t):

m = t[0]

for x in t:

if x[1] >= m[1]:

m = x

return m

L = [('Alice', 17), ('Barnabé', 17),('Casimir', 17), ('Doriane', 17),('Emilien', 14), ('Fabienne', 16)]

Quelle est alors la valeur de maxi(L) ?

Réponses :

***Réponses* :**

A- ('Alice',17)

B- ('Doriane',17)

C- ('Fabienne',17)

D- ('Emilien',14)

**Thème E : architectures matérielles et systèmes d'exploitation**

**Question E.1**

Quel est l'élément qui ne fait pas partie de l'architecture du modèle de Von Neumann ?

***Réponses* :**

A- l'unité d'entrée

B- l'unité arithmétique et logique

C- la mémoire centrale

D- l'unité d'affichage

**Question E.2**

Que peut-on dire du système de fichiers, suite à l’exécution des commandes suivantes ?

% ls

entier.py flottant.py readme.md

% mkdir foo

% mv \*.py foo

***Réponses* :**

A- Les fichiers entier.py, flottant.py, et foo ont été déplacés dans le répertoire de l’utilisateur.

B- L’utilisateur foo est propriétaire des fichiers entier.py et flottant.py.

C- Le répertoire foo contient le résultat de l’exécution des deux fichiers entier.py et flottant.py.

D- Le répertoire foo contient deux fichiers d’extension .py.

**Question E.3**

Dans un ordinateur, que permet de faire la mémoire vive ?

***Réponses* :**

A- Stocker les données de façon permanente

B- Afficher les informations sur l’écran

C- Réaliser les calculs

D- Stocker les données de façon temporaire

**Question E.4**

Quelle est la commande qui permet d’afficher le nom du répertoire courant dans le shell GNU/Linux ?

***Réponses* :**

A- mkdir

B- pwd

C- cd

D- ls -l

**Thème F : langages et programmation**

**Question F.1**

Pour quelles valeurs booléennes des variables a, b et c l'expression (a or b) and (not c) a-t-elle pour valeur True ?

***Réponses* :**

A- a = True b = False c = True

B- a = True b = False c = False

C- a = False b = False c = True

D- a = False b = True c = True

**Question F.2**

Ce programme ne renvoie pas toujours ses trois arguments dans l’ordre croissant. Parmi les tests suivants, lequel va permettre de détecter l’erreur ?

def ranger(a, b, c):

if a > b :

a, b = b, a

if b > c:

b, c = c, b

return a, b, c

***Réponses* :**

A- ranger(1, 2, 3)

B- ranger(3, 4, 1)

C- ranger(1, 3, 2)

D- ranger(4, 2, 3)

**Question F.3**

On exécute le script suivant :

a = 4

b = 4

c = 4

while a < 5:

a = a - 1

b = b + 1

c = c \* b

Que peut-on dire ?

***Réponses* :**

A- Ce programme ne termine pas.

B- A la fin de l'exécution, la variable a vaut 5.

C- A la fin de l'exécution, la variable b vaut 34.

D- A la fin de l'exécution, la variable c vaut 42.

**Question F.4**

Soit T un tableau de flottants, a et b deux entiers. On considère une fonction nommée somme renvoyant la somme des éléments du tableau d'indice compris entre a et b définie par :

def somme(T, a, b):

S = 0

for i in range(a, b+1):

S = S + T[i]

return S

Quel ensemble de préconditions doit-on prévoir pour cette fonction ?

***Réponses* :**

A- a < b

B- a < longueur(T)

C- a <= b < longueur(T)

D- a <= b < longueur(T) et T est un tableau trié

**Thème G : algorithmique**

**Question G.1**

Soit L une liste de n nombres réels (n entier naturel non nul). On considère l'algorithme suivant, en langage Python, calculant la moyenne des éléments de L

m = 0

for k in range(n):

m = m + L[k]

m = m/n

Si le nombre n de données double alors le temps d'exécution de ce script :

***Réponses* :**

A- reste le même

B- double aussi

C- est multiplié par n

D- est multiplié par 4

**Question G.2**

Quelle est la valeur de X/m à la fin de l'exécution du code suivant :

L = [1, 2, 3, 4, 1, 2, 3, 4, 0, 2]

x = 0

m = 0

for k in L:

x = x + k

m = m + 1

***Réponses* :**

A- 2

B- 2.2

C- 10

D- 22

**Question G.3**

Un algorithme de recherche dichotomique dans une liste triée de taille n nécessite, dans le pire des cas, exactement k comparaisons. ?

Combien cet algorithme va-t-il utiliser, dans le pire des cas, de comparaisons sur une liste de taille 2n ?

***Réponses* :**

A- k

B- k+1

C- 2k

D- 2k+1

**Question G.4**

La fonction ci-dessous renvoie le maximum d'une liste.

def maximum(L):

m = L[0]

for i in range(1,len(L)):

#

if L[i] > m:

m = L[i]

return m

Au passage dans la ligne marquée d'un #, quelle propriété reste toujours vérifiée ?

***Réponses* :**

A- m est le maximum des éléments L[k] pour i <= k < len(L).

B- m est le maximum des éléments L[k] pour i < k < len(L).

C- m est le maximum des éléments L[k] pour 0 <= k < i.

D- m est le maximum des éléments L[k] pour 0 <= k <= i.