






NUMÉRIQUE ET SCIENCES INFORMATIQUES

SUJET « A »

Qui peut utiliser ce sujet de NUMÉRIQUE ET SCIENCES INFORMATIQUES ?

- Profil Violet **OUI** 
- Profil Jaune **OUI** 
- Profil Vert **NON** 

1) Lors de l'exécution d'une fonction récursive :

- A) les appels récursifs sont exécutés simultanément
- B) les appels récursifs forment une « file d'exécution »
- C) les appels récursifs forment une « pile d'exécution »
- D) le premier appel est exécuté et renvoie une valeur, puis le second et ainsi de suite jusqu'au dernier appel correspondant au cas d'arrêt.

2) Qu'affiche le code Python suivant ?

```
def myfonc(n,p) :  
    assert(n>=p),"L'entier p doit être strictement plus petit que l'entier n."  
    if n==p :  
        return 1  
    else :  
        return p*myfonc(n-1,p)  
print(myfonc(12,2))
```

- A) 1
- B) 144
- C) Rien car une assertion erreur est levée
- D) 1024

3) On a enregistré 10000 entiers naturels dans une pile p1. Quel code stocke le deuxième entier enregistré dans p1 dans la variable val ?

```
A)  
for i in range(9998) :  
    val = depiler(p1)  
print(val)
```

```
B)  
for i in range(9999) :  
    val = depiler(p1)  
print(val)
```

```
C)  
for i in range(9997) :  
    depiler(p1)  
val = depiler(p1)  
print(val)
```

```
D)  
depiler(p1)  
val=depiler(p1)  
print(val)
```

4) Quelle affirmation est correcte ?

- A) Une pile est une structure de données primitive ou élémentaire
- B) Dans une pile un élément est directement accessible par l'intermédiaire de sa position
- C) Les documents envoyés par le spouleur pour être imprimés peuvent être structurés par une file
- D) On ne peut pas ajouter d'élément à une pile

5) Laquelle des propositions suivantes est vraie ?

- A) Une liste chaînée est une structure de données hiérarchique
- B) Une liste chaînée est une structure de données récursive
- C) Une liste chaînée est identique à une file
- D) Une liste chaînée est une structure de données arborescente

6) Soit le code Python suivant :

```
T1 = [randint(1,100) for i in range(100)]  
T2 = T1.sort()
```

Après l'exécution de ce code quelle affirmation n'est pas validée ?

- A) La variable T2 contient une copie du tableau T1 trié dans l'ordre croissant
- B) La variable T2 contient None
- C) La variable T1 contient un tableau de 100 entiers
- D) Le tableau T1 est trié dans l'ordre croissant

**7) Tab est le tableau défini par le code Tab = [[i+j for i in range(10)] for j in range(20)].
L'instruction Tab[10:20][5:][3][:5] renvoie :**

- A) un tableau contenant tous les éléments de Tab sauf les éléments d'indice 10 et 20
- B) []
- C) le tableau [18, 19, 20, 21, 22, 23]
- D) le tableau [18, 19, 20, 21, 22]

8) Que fait le code Python suivant ?

```
t1 = [1,3,5,4,5]  
t2 = [0, 3, 4, -9, -6, 7, 8, 9, 10, 11]  
t = t1+ t2[2:8]  
k = 0  
while k>=0:  
    k = t[k]  
print("k = ",k)
```

- A) entre dans une boucle infinie
- B) arrête son exécution avec un message d'erreur affiche
- C) affiche k = -9
- D) affiche k = -6

9) Quelle affirmation n'est pas vérifiée ?

- A) Le système range les éléments d'un tableau dans des cases mémoires contigües
- B) Le système range toujours les éléments d'une liste chaînée dans des cases mémoires contigües
- C) Une liste chaînée est une séquences d'éléments, chacun des éléments contient une valeur et une référence à un éventuel élément suivant
- D) L'accès à un élément d'un tableau se fait en temps constant

10) Dans le dictionnaire $d1 = \{i:\text{str}((2*i+1)\%5) \text{ for } i \text{ in range}(100) \text{ if } i\%4==0\}$:

- A) Il y a 24 éléments
- B) les clés sont des entiers impairs multiples de 5
- C) $d1[80] = '1'$
- D) il n'y a aucune valeur égale à '0'

11) Dans le dictionnaire $d1 = \{i:\text{str}((2*i+1)\%5) \text{ for } i \text{ in range}(100) \text{ if } i\%4==0\}$ le nombre de valeurs distinctes est :

- A) 3
- B) 4
- C) 2
- D) 5

12) En programmation orientée objet en Python, dans la définition d'une classe, la méthode `__init__` :

- A) est obligatoire
- B) est exécutée à chaque instanciation d'un objet
- C) ne prend que le seul paramètre `self`
- D) doit être définie avant toutes autres méthodes de la classe

13) On exécute le programme Python suivant :

```
class Pneu():
    def __init__(self,l=205,h=60,r="D",d=16):
        self.largeur = l
        self.hauteur = h
        self.structure = r
        self.diametre = d
    def getAttributs(self):
        return self.largeur, self.hauteur, self.structure, self.diametre
    def setLargeur(self,l):
        self.largeur = l
    def setHauteur(self,h):
        self.hauteur = h
    def setStructure(self,r):
        self.structure = r
    def setDiametre(self,d):
        self.diametre = d
p1 = Pneu(205,55,"D",16)
```

Quelle proposition est vraie ?

- A) `p1` contient le tuple `(205,60,"D",16)`
- B) `p1` contient le tuple `(205,55,"D",16)`
- C) `p1` contient le dictionnaire `{"largeur" : 205, "hauteur" : 60, "structure" : "D", "diametre" : 16}`
- D) `p1` contient une référence à un objet de la classe `Pneu`

14) La classe Pneu a été définie à la question précédente. On exécute les instructions suivantes :

```
p2 = Pneu()  
p3 = p2  
p3.setHauteur(55)  
print([p2.getAttributs()]])
```

Quel affichage a-t-on à l'écran ?

- A) [(205,55,'D',16)]
- B) (205,60,'D',16)
- C) [205,60,'D',16]
- D) [205,55,'D',16]

15) La classe Pneu a été définie à la question précédente. On exécute les instructions suivantes :

```
p4 = Pneu(205,75,"R",20)  
p4.prix = 79.9  
print((p4.getAttributs()[0],p4.getAttributs()[1],p4.getAttributs()[2],p4.getAttributs()[3],p4.prix))
```

- A) le message d'erreur AttributeError: 'Pneu' object has no attribute 'prix' est affiché
- B) l'attribut 'prix' est partagé avec tous les objets de la classe Pneu
- C) l'affichage produit est le tuple (205, 75, 'R', 20, 79.9)
- D) l'affichage produit est le tuple (205, 75, 'R', 20, None)

16) Le concept de la programmation orientée objet :

- A) a été introduit au début de la décennie 1960
- B) a été introduit au début de la décennie 1970
- C) a été initié par *Guido van Rossum*
- D) a été initié par *James Gosling* et *Patrick Naughton*

17) Dans un réseau local une machine M1 à l'adresse IP 192.168.2.57 et le masque de sous-réseau est 255.255.255.0

- A) la machine M2 dont l'adresse IP est 192.168.3.58 est dans le même réseau local que M1
- B) 192.168.2.57 n'est pas une adresse IP attribuable à la machine M3 de ce réseau local
- C) aucune machine de ce réseau ne peut avoir 192.168.2.256 pour adresse IP
- D) l'adresse de ce réseau local est obtenue en appliquant l'opérateur ET binaire entre l'adresse IP de M1 et le masque de sous-réseau

18) Quelle proposition est vraie parmi les quatre suivantes ?

- A) la notation 195.83.224.96/20 signifie que l'adresse 195.83.224.96 est l'IP d'une machine appartenant à un sous-réseau contenant 20 machines
- B) la notation 195.83.224.96/20 signifie que l'adresse 195.83.224.96 est l'IP d'une machine appartenant à un réseau subdivisé en 20 sous-réseaux
- C) la notation 195.83.224.96/20 signifie qu'il ne reste plus que 20 adresses IP possibles pour adresser de nouvelles machines du sous-réseau.
- D) le masque de réseau contient que des 1 sur les 20 premiers bits de poids fort

19) Combien de machines peut-on adresser dans le sous-réseau 185.100.230.0/20 ?

- A) 1022
- B) 1024
- C) 4094
- D) 4096

20) Une fonction récursive terminale est :

- A) une fonction récursive qui se termine dans tous les cas
- B) une fonction récursive qui s'appelle elle-même comme dernière opération
- C) une fonction récursive qui n'a pas besoin de cas d'arrêt
- D) une fonction qui n'utilisent pas de variables locales

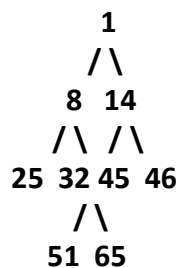
21) Dans un arbre binaire de profondeur 10, on suppose que la racine est à la hauteur 0, le nombre maximum de nœuds est égal à :

- A) 2047
- B) 1024
- C) 2048
- D) 1023

22) On admet que la racine d'un arbre binaire a une hauteur égale à 0. Quelle est la hauteur d'un arbre binaire parfait ayant n nœuds ?

- A) $n+1$
- B) $\log_2(n+1)-1$
- C) $\log_2(n)-1$
- D) n^2

23) On considère l'arbre binaire représenté ci-dessous :



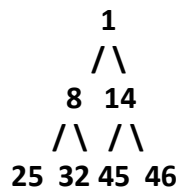
Cet arbre correspond à :

- A) un arbre binaire de recherche
- B) un arbre binaire parfait
- C) une suite des valeurs obtenues dans un parcours profondeur préfixe est 25,51,65,32,8,45,46,14,1
- D) une suite des valeurs obtenues dans un parcours profondeur infixe est 25,8,51,32,65,1,45,14,46

24) La classe Node suivante est utilisée pour implémenter un arbre binaire en Python :

```
class Node():
    def __init__(self, val):
        self.left = None
        self.right = None
        self.value = val
```

Quel code Python implémente l'arbre binaire représenté par la figure ci-dessous en utilisant la classe Node ?



A)

```

arb = Node(1)
arb.left=Node(8)
arb.left.left=Node(25)
arb.left.right=Node(32)
arb.right=Node(14)
arb.right.left=Node(45)
arb.right.right=Node(46)
    
```

B)

```

arb = Node(1,Node(8,Node(25),Node(32)),Node(14,Node(45),Node(46)))
    
```

C)

```

arb = Node()
arb.value = 1
arb.left.value=Node(8)
arb.left.left.value=Node(25)
arb.left.right.value=Node(32)
arb.right.value=Node(14)
arb.right.left.value=Node(45)
arb.right.right.value=Node(46)
    
```

D)

```

arb = (1,(8,(25,None,None),(32,None,None)),(14,(45,None,None),(46,None,None)))
    
```

25) On a implémenté en Python un graphe par la matrice d'adjacence suivante :

$M1 = [[0, 1, 0, 1], [1, 0, 1, 0], [0, 1, 0, 1], [1, 0, 1, 0]]$

Combien d'arêtes ce graphe possède-t-il ?

A) 8

B) 5

C) 4

D) 16

26) La fonction mystère ci-dessous prend en paramètre une matrice m d'adjacence d'un graphe non orienté.

```

def mystere(m):
    var = 0
    for i in range(len(m)):
        for j in range(i + 1, len(m)):
            if m[i][j] == 1:
                var += 1
    return var
    
```

Que renvoie cette fonction quand elle prend en argument la matrice $M2 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$?

- A) 8
- B) 5
- C) 4
- D) 16

27) Pourquoi est-il préférable d'utiliser un dictionnaire d'adjacence plutôt qu'une matrice d'adjacence pour implémenter un graphe ?

- A) parce qu'il est plus facile de parcourir le graphe avec un dictionnaire d'adjacence
- B) parce que le dictionnaire d'adjacence permet de mieux visualiser le graphe
- C) parce qu'une matrice permet de connaître plus facilement le nombre de sommet du graphe
- D) parce qu'une matrice occupe trop de place en mémoire pour les graphes avec peu d'arêtes

28) Quelle fonction renvoie le dictionnaire d'adjacence du graphe dont la matrice m est la matrice d'adjacence ?

A)

```
def amyster1(m):  
    """m est la matrice d'adjacence d'un graphe"""  
    var = {}  
    for i in range(len(m)):  
        l = []  
        for j in range(len(m)):  
            if m[i][j] == 1:  
                l.append(j)  
        var[i]=l  
    return var
```

B)

```
def amyster2(m):  
    """m est la matrice d'adjacence d'un graphe"""  
    var = {}  
    for i in range(len(m)):  
        l = []  
        for j in range(i+1,len(m)):  
            if m[i][j] == 1:  
                l.append(j)  
        var[i]=l  
    return var
```

C)

```
def amyster3(m):  
    """m est la matrice d'adjacence d'un graphe"""  
    var = {}  
    for i in range(len(m)):  
        for j in range(i+1,len(m)):  
            l = []  
            if m[i][j] == 1:  
                l.append(j)  
        var[i]=l  
    return var
```


D)
def amyster4(m):
 """m est la matrice d'adjacence d'un graphe"""
 var = {}
 for i in range(len(m)):
 l=[]
 for j in range(len(m)):
 if m[i][j] == 1:
 l.append(j)
 var[i]=l
 return var

29) Qui est connu comme étant à l'origine de la théorie des graphes ?

- A) Alan Turing
- B) John von Neumann
- C) Leonard Euler
- D) Kurt Gödel

30) Quel est le principal avantage des SOC, systèmes sur puce, par rapport aux systèmes traditionnels ?

- A) Performances inférieures
- B) Nécessité de plusieurs composants externes
- C) Portabilité limitée
- D) Consommation d'énergie réduite et taille compacte

31) On exécute le code Python suivant :

```
def melange(g, d):  
    r=[]  
    i=j=0  
    while i<len(g) and j<len(d):  
        if g[i]<d[j]:  
            r.append(g[i])  
            i=i+1  
        else:  
            r.append(d[j])  
            j=j+1  
    r=r+g[i:]  
    r=r+d[j:]  
    return r  
print(melange([2, 5, 8, 10, 15],[0, 1, 10, 15, 19, 20]))
```

Quel est l'affichage obtenu ?

- A) [0, 1, 2, 5, 8, 10, 15, 19, 20]
- B) [0, 1, 2, 5, 8, 10, 10, 15, 15, 19, 20]
- C) [2, 5, 8, 10, 15, 0, 1, 10, 15, 19, 20]
- D) [2, 5, 8, 10, 15, 0, 1, 10, 19, 20]

32) Que fait la commande Linux kill -9 760 ?

- A) elle force l'arrêt du processus dont le PID est 760
- B) elle fait redémarrer le processus dont le PID est 760 après une pause correspondant à 9 quantum
- C) elle arrête le processus dont le PID est 760 après 9 cycles d'exécution
- D) elle met le processus dont le PID est 760 en pause pendant 9 millisecondes

33) Quelle instruction SQL est utilisée pour créer une contrainte d'unicité sur un attribut ?

- A) CONSTRAINT UNIQUE
- B) UNIQUE
- C) UNIQUE KEY
- D) CHECK UNIQUE

34) Quelle est la différence entre WHERE et HAVING ?

- A) Aucune, WHERE et HAVING sont interchangeables
- B) HAVING filtre avant l'agrégation, WHERE après
- C) WHERE est toujours suivi d'une expression booléenne et pas HAVING
- D) WHERE filtre avant l'agrégation, HAVING après

35) Voici le schéma relationnel d'une base de données :

- Achete(#idFilm, #idUtil, aPaye)
- Utilisateur(idUtil, nom, prenom, tel)
- Film(idFilm, titre, #idGenre, prix)
- Genre(idGenre, nom)

Comment récupérer la liste des films et leur genre associé ?

- A) SELECT titre, nom FROM Film;
- B) SELECT titre, nom FROM Film JOIN Genre;
- C) SELECT titre, Genre.nom FROM Film JOIN Genre ON Film.idGenre = Genre.idGenre;
- D) SELECT Film.nom, Genre.titre FROM Film, Genre;

36) Voici le schéma relationnel d'une base de données :

- Achete(#idFilm, #idUtil, aPaye)
- Utilisateur(idUtil, nom, prenom, tel)
- Film(idFilm, titre, #idGenre, prix)
- Genre(idGenre, nom)

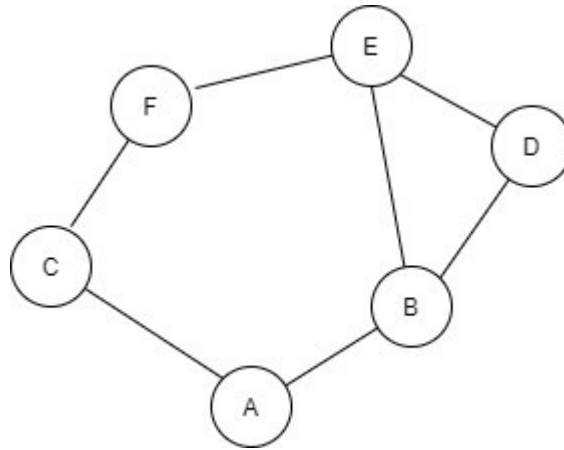
Quelle requête affiche le nom et prénom des utilisateurs ayant acheté un film à plus de 15 ?

- A) SELECT nom, prenom FROM Utilisateur WHERE prix > 15;
- B) SELECT nom, prenom FROM Utilisateur INNER JOIN Achete ON Utilisateur.idUtil = Achete.idUtil INNER JOIN Film ON Achete.idFilm = Film.idFilm WHERE Film.prix > 15;
- C) SELECT nom, prenom FROM Utilisateur WHERE Film.prix > 15;
- D) SELECT nom, prenom FROM Film INNER JOIN Utilisateur ON prix > 15;

37) Pour sélectionner tous les films avec leur prix total payé par les utilisateurs, on utilise :

- A) SELECT Film.titre, SUM(Film.prix) FROM Achete JOIN Film ON Achete.idFilm = Film.idFilm GROUP BY Film.titre;
- B) SELECT titre, SUM(prix) FROM Film;
- C) SELECT titre, prix FROM Achete;
- D) SELECT titre, prix FROM Film NATURAL JOIN Achete;

38) On représente un graphe par la figure suivante :



Quel est le degré du sommet 'A' dans ce graphe ?

- A) 2
- B) 1
- C) 4
- D) 3

39) On considère le code Python suivant :

```

def p (G, S, l=None):
    if l is None:
        l = []
    l.append(S)
    for voisin in G[S]:
        if voisin not in l:
            p(G,voisin,l)
    return l
G1 = {
    'A': ['B', 'C'],
    'B': ['A', 'D', 'E'],
    'C': ['A', 'F'],
    'D': ['B', 'E'],
    'E': ['B', 'F'],
    'F': ['C', 'E']
}
    
```

Que se passera-t-il si on exécute l'instruction `p(G1,'X')` ?

- A) La fonction renvoie une liste vide
- B) La fonction renvoie None
- C) La fonction renvoie une erreur `KeyError`
- D) La fonction va boucler indéfiniment

40) Grace Hopper est considérée comme une pionnière dans le domaine des langages de programmation

- A) parce qu'elle a inventé les bases de l'intelligence artificielle
- B) parce qu'elle a inventé le langage Fortran
- C) parce qu'elle a inventé le langage assembleur
- D) parce qu'elle a conçu le premier compilateur, qui traduit le code en langage machine

• • • FIN • • •

Ce sujet est la propriété intellectuelle exclusive du Concours Avenir. Il ne doit en aucun cas être emporté par les candidats à la fin de l'épreuve. Il doit être rendu à l'équipe surveillante en même temps que sa grille réponses associée.