14.01.20 # Épreuve de bac blanc NSI Première 2h

N° candidat

Notation

- + 3 pts par réponse correcte
- - 1 pt par réponse fausse
- 0 pour une absence de réponse ou une réponse multiple
- Si sur chacune des quatre parties, le total des points obtenus est négatif, son résultat est évalué à 0.

Nombre pts :

$$Note = 20 \times \frac{Nombre\ points}{3 \times 42}$$

Note =

Pour chaque question, il n'y a qu'une seule bonne réponse

1ère Partie : Représentation des données : types et valeurs de base (11 questions)

Question 1 : Quel est le nombre minimal de bits nécessaire pour représenter le nombre décimal 227 ?

☑ 8
☐ 11
☐ 7
☐ 6

Question 2 : À quelle valeur binaire correspond le nombre noté A3 en hexadécimal ?

☑ 10100011
☐ 10000011
☐ 10001011
☐ 10110011

Question 3 : Quelle somme de nombre binaire est égale à 51 ?

☐ 100010 + 001101
☑ 100110 + 001101
☐ 000010 + 001101
☐ 100110 + 011101

Question 4 : La représentation binaire de 17 s'écrit :

☑ 0001 0001
☐ 0001 0111
☐ 0010 0111
☐ 0010 0111

Question 5 : La représentation binaire de 3,625 s'écrit :

☐ 11,111
☑ 11,101
☐ 11,001
☐ 11,011

Question 6 : La chaîne de caractères ch a pour valeur `'123456789'`. Quelle est la valeur de l'expression ch[7] + ch[8] ?

- ☐ `15`
- ☐ `17`
- ☐ `'78'`
- ☑ `'89'`

Question 7 : La représentation de -13 sur 5 bits s'écrit :

- ☐ `01011`
- ☐ `10111`
- ☑ `10011`
- ☐ `00111`

Question 8 : La variable x contient la valeur 3 et la variable y contient la valeur 4. Parmi les expressions suivantes, laquelle renvoie True ?

- ☑ `(x == 3) or (y == 5)`
- ☐ `(x == 3) and (y == 5)`
- ☐ `(x != 3) or (y == 5)`
- ☐ `y < 4`

Question 9 : Sur combien d'octets a été codé la chaîne de caractères suivante en utf-8 : `''`*égalités*`''`?

- ☑ `10`
- ☐ `8`
- ☐ `32`
- ☐ `1`

Question 10 : Quelle expression a pour table de vérité :

a	b	Expression
0	0	0
0	1	1
1	0	1
1	1	0

- ☐ `(a and b) or (not(a) and b)`
- ☐ `(a and not(b)) and (not(a) and b)`
- ☑ `(a and not(b)) or (not(a) and b)`
- ☐ `(a or not(b)) and (not(a) or b)`

Question 11 : L'expression `0.2+0.1>0.3` a la valeur True. Quelle en est la raison ?

- ☐ C'est une erreur de la machine.
- ☑ C'est parce que 0.1 ne peut pas être représenté en flottant de manière exacte.
- ☐ C'est parce que la machine n'utilise que 32 bits pour coder les flottants.
- ☐ C'est parce que > signifie >= pour l'interpréteur Python.

2ième Partie : Représentation des données : types construits (10 questions)

Question 12 : On considère la liste suivante :
`li=[9, 7, 8, 6]`
Quelle expression parmi les suivantes est vraie ?

- ☑ `li[2] == 8`
- ☐ `li[1] == 9`
- ☐ `li[0] == 0`
- ☐ `li[0] != 9`

Question 13 : On considère le code ci-dessous, qu'obtient-on à l'exécution ?

```
li=["louis",15,(0,7)]
print(type(li[2]))
```

- ☐ `<class 'int'>`
- ☐ `<class 'list'>`
- ☐ `<class 'str'>`
- ☑ `<class 'tuple'>`

Question 14 : Soit la liste `li=[[1,'b','c'],'d']`, quelle expression renvoie True ?

- ☑ `'b'==li[0][1]`
- ☐ `'b'==li[0][0]`
- ☐ `'b'==li[1][0]`
- ☐ `'b'==li[0][2]`

Question 15 : Que renvoie le code suivant à l'exécution ?

```
li=[i%3 for i in range(6)]
print(li)
```

- ☐ `[0, 1, 2, 3, 4, 5]`
- ☑ `[0, 1, 2, 0, 1, 2]`
- ☐ `[0, 3, 6, 9, 12, 15]`
- ☐ `[3, 4, 5, 6, 7, 8]`

Question 16 : On définit une matrice A à l'aide d'une liste de listes par :

```
A=[]
for ligne in range(2):
        B=[]
        for colonne in range(3):
                B.append(ligne-colonne)
        A.append(B)
```

Combien d'éléments contient la matrice A ?

- ☑ 6
- ☐ 2
- ☐ 3
- ☐ 5

Question 17 : Lequel de ces objets est un dictionnaire ?

☑ `{'a':3 , 'b':7 , 'c':5}`
☐ `[3,7,2,1]`
☐ `(3,2,1,4)`
☐ `([3,2],[7,2],[8,1])`

Question 18 : Soit le tableau défini de la manière suivante :
`tableau=[4,5,3,7,8]`
L'instruction `tableau[3]=4` va permettre de :

☑ remplacer le 7 par un 4 dans ce tableau
☐ est impossible (non autorisée)
☐ remplacer le 3ème élément par 4
☐ renvoyer False (faux) si `tableau[3]` ne contient pas 4

Question 19 : Soit le tuple défini de la façon suivante :

`tuple = (4,5,3,7,8)`
L'instruction `tuple[3] = 4` va permettre de :

☐ remplace par 4 l'élément d'indice 3
☐ renvoyer False (faux) si `tuple[3]` ne contient pas 4
☐ remplace le 3ème élément par 4
☑ est impossible (non autorisée)

Question 20 :

```
panier={'pomme':5,
        'poire':2,
        'banane':7,
        'orange':4}
```

On souhaite connaître le nombre de fruits dans le panier et on donne le code suivant, que faut-il utiliser à la place des points de suspension ?

```
def nombre_de_fruits(d):
    compteur=0
    for fruits in ... :
        compteur = compteur + d[fruits]
    return compteur
```

☑ `d.keys()`
☐ `d.values()`
☐ `d.items()`
☐ `d.index()`

Question 21 : Voici une fonction écrite en langage Python qui prend en paramètres un dictionnaire dico et une chaîne de caractères chaine.

```
def  affiche(dico, chaine):
    i=0
    for element in dico.keys():
        if dico[element]==chaine:
            i=i+1
    return i
```

☑ Cette fonction compte le nombre d'apparitions de chaine parmi les valeurs du dictionnaire dico.
☐ Cette fonction renvoie True si chaine fait partie des valeurs du dictionnaire dico.
☐ Cette fonction renvoie toujours 0.
☐ Cette fonction renvoie le nombre de valeurs du dictionnaire dico.

3ième Partie : Traitement des données en tables (10 questions)

Question 22 : Soit le code :

```
table=[["Amiens",3.1,6.8],["Lille",5.5,6],["Paris",-1,2.7]]
Qu'affiche l'instruction print(table[1][0]) ?
```

☐ `3.1`
☑ `"Lille"`
☐ `"Amiens"`
☐ `5.5`

Question 23 : Le code :

```
fichier = open('chicon.csv', encoding='utf-8')
ma_table = list(csv.DictReader(fichier, delimiter =";"))
```

a permis d'importer les données des 4 colonnes d'un fichier "chicon.csv". Chaque élément de la liste étant un dictionnaire comportant 4 clés.
Parmi les contenus de ces 4 fichiers, quel est celui qui pourra être lu convenablement ?

☑
```
1;1;8;55
3;1;21;23
```

☐
```
1 1 8 55
3 1 21 23
```

☐
```
1,1,8,55
3,1,21,23
```

☐
```
1|1|8|55
3|1|21|23
```

Question 24 : Un fichier CSV contient la liste des concurrents à une course à pied.
La structure de ce fichier CSV est la suivante :
`Prénoms,Noms,Sexes`
Un fois importées en python, les données du fichier sont stockées dans une variable `table`.

```
table = [(1, "gaston", "lagaffe", "M"),
        (2, "jessie ", "jane", "F")]
```

Quel est le codage python qui permet d'afficher la liste de tous les concurrents hommes ("M") ?

☑
```
for c in table :
      if c[3] == "M" :
            print(c)
```

☐
```
for c in table :
      if c["M"] == 3 :
            print(c)
```

☐
```
for c in table :
      if c == "M"[4] :
            print(c)
```

☐
```
for c in table :
      if "M" == c[4] :
            print(c)
```

Question 25 : On a récupéré le contenu d'un fichier csv contenant le nom le prénom et l'age de personnes dans une table implémentée par la liste de dictionnaires

```
table = [{"nom": "dupont","prenom": "jean","age": 16}, {"nom": "durant",
"prenom": "pierre", "age": 15}, .... {"nom": "doe","prenom": "jane","age": 16}]
```

Quelle syntaxe permet de récupérer la liste des noms des personnes de 16 ans

☐ `[personne[nom] for personne in table if personne[age]==16]`

☐ `[personne["nom"] if personne["age"]==16 for personne in table ]`

☑ `[personne["nom"] for personne in table if personne["age"]==16]`

☐ `[nom if age==16 for nom,age in table]`

Question 26 : on dispose de deux tables

```
t1=[{"id":1,"nom":"lui"},{"id":2,"nom":"elle"},{"id":3,"nom":"etlui"},{"id":4,"
nom":"etelle"}]
t2=[{"id":1,"age":14},{"id":4,"age":15}]
```

On veut ajouter les âges aux données de la table t1 pour former la table t0, quel code le permettrait ?

☑
```
t0=[]
for enregistrement1 in t1:
      for enregistrement2 in t2:
            if enregistrement1["id"] == enregistrement2["id"]:
                  enregistrement1["age"]=enregistrement2["age"]
                  t0.append(enregistrement1)
```

☐
```
t0=[]
for enregistrement in t2:
     if enregistrement[id] in t1:
            t0[enregistrement["id"]]=enregistrement["age"]
```

☐
```
t0=[]
for enregistrement1 in t2:
    for enregistrement2 in t1:
        if enregistrement1[id] == enregistrement2[id]:
            t2[enregistrement1["id"]]=enregistrement2["age"]
```

☐
```
t0=[]
for enregistrement1 in t1:
    for enregistrement2 in t2:
        if enregistrement1["id"] == enregistrement2["id"]:
            enregistrement1["age"]=enregistrement2["age"]
            t0+={enregistrement1}
```

Question 27 : À partir d'un fichier .csv, on obtient les informations suivantes (nom de la ville, température minimale et la température maximale):
`li=[['Bordeaux',14,22],['Lille',10,16],['Marseille',15,25],['Paris',14,20]]`
Quelle instruction permettrait d'obtenir la température minimale à Marseille?

☐ `li[3][2]`
☐ `li[2][3]`
☑ `li[2][1]`
☐ `li[1][2]`

Question 28 : On dispose d'un fichier de données csv nommé fic.csv que l'on veut ouvrir en mode lecture avec Python pour récupérer ces données. Quelle instruction est correcte ?

☐ `f.open("fic.csv","r")`
☐ `f.open("fic.csv","read")`
☑ `f=open("fic.csv","r")`
☐ `f=open("fic.csv","read")`

Question 29 : Pour suivre la propagation des épidémies, de nombreuses données sont recueillies par les institutions internationales comme l'O.M.S. Par exemple, pour le paludisme, on dispose de la table *palu* qui recense le nombre de nouveaux cas confirmés et le nombre de décès liés au paludisme ; certaines lignes de cette table sont données en exemple (on précise que Iso est un identifiant unique pour chaque pays) :

Nom	Iso	Année	Cas	deces
Bresil	BR	2009	309316	85
Bresil	BR	2010	334667	76
Kenya	KE	2010	898531	26017
Mali	ML	2011	307035	2128
….	….	….	….	….

Cette table peut être représentée par une liste de listes
palu=[['Bresil','BR',2009,309316,85],
['Bresil','BR',2010,334667,76],
['Kenya','KE',2010,898531,26017],
['Mali','ML',2011,307035,2128]]
La table *demographies* recense la population totale de chaque pays ; certaines lignes de cette table sont données en exemple :

Pays	Periode	pop
BR	2009	193020000
BR	2010	194946000
KE	2011	33987000
….	….	….

Cette table peut être représentée par une liste de listes appelée *demographies* :
demographies=[['BR',2009,193020000],
['BR',2010,194946000],
['KE',2010,40909000],
['ML',2011,33987000]].
Quel script python permet d'extraire le nom des pays référencés dans la table *palu* et de les classer par ordre alphabétique ? (on rappelle que la méthode .sort() trie la liste à laquelle elle s'applique.)

☐ `pays=[]`

```
for i in palu:
    if i not in pays:
        pays.append(i[0])
pays.sort()
print(pays)
```

☐
```
pays=[]
for i in palu:
    if i not in pays:
        pays.append(i)
pays.sort()
print(pays)
```

☑
```
pays=[]
for i in palu:
    if i[0] not in pays:
        pays.append(i[0])
pays.sort()
print(pays)
```

☐
```
pays=[]
for i in palu:
    if i in pays:
        pays.append(i[0])
pays.sort()
print(pays)
```

Question 30 : Cette question est indépendante de la question 29 mais reprend le même contexte. On souhaite afficher la liste palu ordonnée par pourcentage croissant de décès par cas présenté. Quel script python faut-il retenir ? Aucune connaissance particulière concernant l'écriture de l'instruction utilisée n'est requise.

☑ `print(sorted(palu, key=lambda effectif: 100*effectif[4]/effectif[3]))`
☐ `print(sorted(palu, key=lambda effectif: 100*effectif[3]/effectif[4]))`
☐ `print(sorted(palu, key=lambda effectif: 100*effectif[2]/effectif[4]))`
☐ `print(sorted(palu, key=lambda effectif: 100*effectif[3]/effectif[2]))`

Question 31 : Cette question est indépendante de la question 29 mais reprend le même contexte.
Quel script permet d'obtenir la table *repartition* définie par :
`repartitions=[['nom',annee,cas,pop],...] ,`

[['Bresil', 2009, 309316, 193020000],
['Bresil', 2010, 334667, 194946000],
['Kenya', 2010, 898531, 40909000],
['Mali', 2011, 307035, 33987000]]?

☐
```
repartitions=[]
for i in palu:
    for j in demographies:
        if i[1]==j[1] and i[2]==j[0]:
            repartitions.append([i[1],i[2],i[3],j[2]])
print(repartitions)
```

☐
```
repartitions=[]
for i in palu:
     for j in demographies:
          if i[2]==j[1] and i[2]==j[1]:
               repartitions.append([i[1],i[1],i[3],j[2]])
          print(repartitions)
```

☑
```
repartitions=[]
for i in palu:
    for j in demographies:
        if i[1]==j[0] and i[2]==j[1]:
            repartitions.append([i[0],i[2],i[3],j[2]])
print(repartitions)
```

☐
```
repartitions=[]
for i in palu:
    for j in demographies:
        if i[3]==j[1] and i[2]==j[3]:
            repartitions.append([i[1],i[1],i[2],j[2]])
print(repartitions)
```

4ième Partie : Algorithmique et programmation (11 questions)

Question 32 : Que renvoie `mystere(14,5)`, la fonction mystère étant définie par :

```
def mystere(a,b):
    """
    : param a : (int)
    : param b : (int)
    : CU : b non nul
    """
    r=a
    q=0
    while r>=b:
        r=r-b
        q=q+1
    return q,r
```

- ☐ `(3, 1)`
- ☐ `(4, 2)`
- ☑ `(2, 4)`
- ☐ `(1, 3)`

Question 33 : Qu'obtient-on quand on exécute le programme suivant :

```
def f(x):
    return -1*x+2
liste1=list(range(-6,10))
liste2=[f(u) for u in liste1]
print(liste2)
```

- ☐ `[8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -3, -4, -5, -6, -7, -8]`
- ☑ `[8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -3, -4, -5, -6, -7]`
- ☐ `[8, 7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -3, -4, -5, -6]`
- ☐ `[7, 6, 5, 4, 3, 2, 1, 0, -1, -2, -3, -4, -5, -6]`

Question 34 : Soit le code suivant, qu'obtient-on quand on l'exécute ?

```
matrice=[]
for n in range(4):
    ligne=[4*n+i for i in range(1,5)]
    matrice.append(ligne)
print(matrice)
```

- ☐ `[[1, 2], [5, 6], [9, 10]]`
- ☐ `[[1, 2, 3, 4, 5], [5, 6, 7, 8, 9], [9, 10, 11, 12, 13], [13, 14, 15, 16, 17]]`
- ☑ `[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]]`
- ☐ `[[1, 2, 3], [5, 6, 7], [9, 10, 11], [13, 14, 15], [17, 18, 19]]`

Question 35 : Soit la fonction mystere définie ci-dessous.
Que renvoie `mystere([2,5,6,8],[1,4,7,8,9])` ?

```
def mystere(liste1,liste2):
    liste=[]
    i,j=0,0
    while i<len(liste1) and j<len(liste2):
        if liste1[i]<liste2[j]:
            liste.append(liste1[i])
            i=i+1
        else:
            liste.append(liste2[j])
            j=j+1
    return liste
```

☐ `[2, 2, 5, 5, 6, 8, 8, 8]`
☐ `[1, 2, 7, 6, 8, 8, 8]`
☐ `[1, 2, 4, 5, 6, 7, 8, 9]`
☑ `[1, 2, 4, 5, 6, 7, 8, 8]`

Question 36: Soit le code suivant, qu'obtient-on quand on l'exécute ?

```
for x in range(10):
    if x % 3 == 0 or x >= 5:
        print(x, end = ' ')
```

☐ `1 2 4 6 9`
☐ `0 3 5 7 8`
☑ `0 3 5 6 7 8 9`
☐ `1 2 4`

Question 37 : Que renvoie `mystere(3)` si la fonction mystere, écrite en langage Python est :

```
def mystere(a):
    if a < 7:
        return 2 * a
    else:
        return a - 1
```

☐ `3`
☑ `6`
☐ `2`
☐ `5`

Question 38 : Combien de fois l'instruction x=x+2 va-t-elle être exécutée dans le script suivant :

```
x=4
while(x<10):
    x=x+2
```

☑ `3 fois`
☐ `4 fois`
☐ `1 fois`
☐ `5 fois`

Question 39 : Qu'obtient-on après l'exécution de ce script ?

```
i = 0
j = 2
while j > 0 :
    j = j - 1
    i = i + 1
print(i)
```

☐ 1
☐ 3
☑ 2
☐ 0

Question 40 : On considère la fonction suivante :

```
def mystere(liste) :
    s = 0
    for n in liste:
        s = s + n
    return s/len(liste)
```

Quel est le rôle de cette fonction dont le paramètre d'entrée liste est une liste d'entiers non vide ?

☐ Cette fonction retourne la somme des éléments de la liste
☐ Cette fonction retourne l'inverse de la somme des éléments de la liste
☑ Cette fonction retourne la moyenne des éléments de la liste
☐ Cette fonction retourne une erreur

Question 41 : On considère la fonction suivante qui prend une liste non vide en paramètre.

```
def min(liste) :
    #ligne manquante
    for val in liste:
        if val < ret:
            ret = val
    return ret
```

Par quelle instruction faut-il remplacer la 'ligne manquante' pour que la fonction renvoie la plus petite valeur de la liste passée en paramètre ?

☑ `ret = liste[0]`
☐ `ret = 0`
☐ `ret = 10000000000`
☐ Il n'y a pas besoin d'ajouter de ligne, la fonction donne le résultat voulu tel quel.

Question 42: Pour parcourir entièrement le tableau suivant : `tableau = [5,7,8,9]`, quelle est la syntaxe de la boucle pour que l'on peut utiliser

☐ `for i in range (1, len(tableau)-1 ):`
☐ `for i in range (0, len(tableau)-1 ):`
☐ `for i in range (5, 9):`
☑ `for i in range (0, len(tableau)):`