ITI1520, Examen partiel, Automne 2016, 1h20m

Instructions: À lire attentivement!

1. Ecrivez votre nom et votre numéro d'étudiant sur ce papier dans le coin droit supérieur et écrivez les aussi sur la page Scantron.
2. Il faut soumettre ce papier avec les questions et la page Scantron avec les réponses.
3. Répondez à toutes les questions sur la page Scantron, de préférence avec un crayon pour pouvoir les modifier au besoin (stylo ca va aussi, mais vous ne pourrez pas les modifier). Remplissez bien les cercles pour les réponses, pour la correction automatique.
4. Choisissez seulement une rṕonse pour chaque question. Si vous pensez que plusieurs réponses sont correctes, choisissez la plus correcte/précise.
5. Tous les programmes sont en Python 3.
6. Cet examen est à livres fermés. Aucun livre ou appareil électronique (incluant les calculatrices) n'est permis.
7. Les points alloués à chaque question sont les mêmes.
8. Vous pouvez utiliser le verso des pages pour vos calculs et brouillons.
9. Si vous regardez le questionnaire de l'un de vos voisins, vous serez expulsé de la salle d'examen.

1. Quelle est la valeur de l'expression `-1+2**5`?
   (a) 1 (b) 5 (c) 9 (d) 31 (e) 32

2. Quelle est la valeur de l'expression `4//2 == 2%3`?
   (a) True (b) False (c) 2 (d) No (e) Yes

3. Si `a` et `b` sont de type `int`, l'expression `a < b` est de type:
   (a) `float` (b) `int` (c) `str` (d) `bool` (e) `list`

4. Qu'est-ce que le programme Python suivant va afficher sur l'écran?

```
n = 35
print(n % 4 == 0 or n % 3 == 0)
```

   (a) un message d'erreur
   (b) `True`
   (c) `False or False`
   (d) `False or True`
   (e) `False`

5. Combien d'arguments (paramètres actuels) y a-t-il dans l'appel print de la question anterieure?
   (a) 0 (b) 1 (c) 2 (d) 3 (e) 4

6. Qu'est-ce que le programme Python suivant va afficher sur l'écran?

```
s = 'sing'
s[0] = 'r'
s=s+'s'
print(s)
```

   (a) `sing`
   (b) `ring`
   (c) `rings`
   (d) `sings`
   (e) Rien. Le code donne erreur d'éxecution.

7. Qu'est-ce que le programme Python suivant va afficher sur l'écran?

```
nombres = [1, 2, 3]
nombres[0] = "dix"
print(nombres)
```

   (a) `[1, 2, 3]`
   (b) `1, 2, 3`
   (c) `['dix', 2, 3]`
   (d) `'dix 2 3'`
   (e) Rien. Le code donne une erreur d'éxecution.

8. Qu'est-ce que le programme Python suivant va afficher sur l'écran?

```
s1 = 'good'
s2 = s1
s2 = s2 + '-bye'
print(s1)
```

   (a) `good`
   (b) `bye`
   (c) `-bye`
   (d) `good-bye`
   (e) Rien. Le code donne une erreur d'éxecution.

9. Qu'est-ce que le programme Python suivant va afficher sur l'écran?

```
l1 = ['hello']
l2 = l1
l2[0] = 'bye'
print(l1)
```

(a) `['bye']`
(b) `['hello']`
(c) `['hellobye']`
(d) `['byelo']`
(e) Rien. Le code donne une erreur d'éxecution.

10. Qu'est-ce que le programme Python suivant va afficher sur l'écran?

```
l1 = [1,2,3]
l2 = l1
l2= [0,0,0]
print(l1, l2)
```

(a) `[1, 2, 3] [1, 2, 3]`
(b) `[0, 0, 0] [0, 0, 0]`
(c) `[0, 0, 0] [1, 2, 3]`
(d) `[1, 2, 3] [0, 0, 0]`
(e) aucune des déclarations ci-dessus

11. Considerez la fonction suivante:

```
def heure(h):
    if h == 12:
          return "midi"
    noms = [ "minuit", "un", "deux", "trois", "quatre", "cinq",
              "six", "sept", "huit", "neuf", "dix", "onze" ]
    return noms[h%12]
```

L'appel `heure(17)` retourne
(a) le nombre entier 5
(b) la chaine de caractères `'cinq'`
(c) une liste
(d) Rien. Le code donne une erreur d'éxecution.

12. L'appel `heure(29)` retourne
(a) le nombre entier 5
(b) la chaine de caractères `'cinq'`
(c) une liste
(d) Rien. Le code donne une erreur d'éxecution.

13. Quel est le type des paramètres et du résultat retourné dans la définition de la fonction suivante?

```
def num_chars(ch, w1, w2):
    '''
    Retourne le nombre de fois que le caractere ch
    est trouve dans w1 et w2.

    >>> num_chars('h', 'hello', 'hehe')
    3
    '''
```

(a) `(int)->str,str,str`
(b) `(str,str,str)->float`
(c) `(str,str,str)->int`
(d) `(str,str)->int`

14. Une année bissextile est représentée par un nombre non négatif et divisible par 4. Tout nombre entier non négatif divisible par 100 ne représente une année bissextile que s'il est aussi divisible par 400. Laquelle des expressions booléènnes suivantes est vraie si la variable A se réfère à un nombre entier représentant une année bissextile et fausse dans le cas contraire.

(a) `A>= 0 or A % 4 == 0 or (A % 100 != 0 or A % 400 == 0)`
(b) `A>= 0 and A % 4 == 0 and (A % 100 != 0 or A % 400 != 0)`
(c) `A>= 0 and A % 4 == 0 and (A % 100 == 0 or A % 400 != 0)`
(d) `A>= 0 and A % 4 == 0 or (A % 100 != 0 or A % 400 == 0)`
(e) `A>= 0 and A % 4 == 0 and (A % 100 != 0 or A % 400 == 0)`

15. Lesquelles des déclarations suivantes sont correctes au sujet de la fonction orange (qui prend comme paramètre un nombre entier positif)?

```
def orange(num):
    ''' (int)->int
    Precondition: num est positif '''

    t=1
    for i in range(num):
        t = t * 2
    return t
```

(a) La fonction calcule la valeur 1 à la puissance `num`.
(b) La fonction calcule la valeur 2 à la puissance `num`.
(c) La fonction calcule 2 multiplié par `num`.
(d) La fonction calcule le factoriel de `num`
(e) aucune des déclarations ci-dessus

16. On doit compléter la définition de la fonction suivante:

```
def on_line(a,b,p,q):
    '''(int, int, int, int)->bool
       Retourne True si le point (p,q) est sur la ligne y=ax+b et False sinon
       Precondition: a est un nombre entier positif
    '''
```

On doit compléter la fonction avec:

I)
```
if q == a * p + b:
    return True
else:
    return False
```

II) `return q == a * p + b`

III) `print q == a * p + b`

(a) I et II
(b) I et III
(c) I seulement
(d) II seulement
(e) III seulement

17. Considerez la fonction suivante:

```
def guitar(n):
     '''(int)->None
     Precondition: n est un entier positif
     '''
     for i in range(1,n+1):
          if(n%i == 0):
               print(i, end=" ")
     print()
```

Que fait la fonction?

(a) affiche tous les diviseurs de `n`
(b) affiche tous les diviseurs de `n+1`
(c) affiche tous les entiers premiers inférieurs ou éqaux à `n`
(d) affiche tous les entiers non premiers inférieurs ou éqaux à `n`
(e) affiche tous les entiers divisibles par `i`

18. Qu'est-ce que le programme Python suivant va afficher sur l'écran?

```
def foo1(a, b):
     print(a+b)

x1 = 10 * foo1(5,5)
print(x1)
```

(a) Il affiche 10 et 100.
(b) Il affiche 10 seulement.
(c) Il affiche 100 seulement.
(d) Il affiche 10 et donne une erreur.
(e) Il affiche 250 seulement.

19. Qu'est-ce que le programme Python suivant va afficher sur l'écran?

```
def foo2(a, b):
     return a+b

x2 = 10 * foo2(5,5)
print(x2)
```

(a) Il affiche 10 et 100.
(b) Il affiche 10 seulement.
(c) Il affiche 100 seulement.
(d) Il affiche 10 et donne une erreur.
(e) Il affiche 250 seulement.

20. Qu'est-ce que le programme Python suivant va afficher sur l'écran?

```
def foo3(a,b):
     print(a+b)
     return a+b

x3 = 10 * foo3(5,5)
print(x3)
```

(a) Il affiche 10 et 100.
(b) Il affiche 10 seulement.
(c) Il affiche 100 seulement.
(d) Il affiche 10 et donne une erreur.
(e) Il affiche 250 seulement.

21. Voici une version de la fonction sequenceDesDeux du devoir 3 qui prend une liste de nombres et retourne True sil y a au moins une sq́uence de deux éléments consécutifs égaux, et False dans le cas contraire.

```
def sequenceDesDeux(x):
     '''(list)-> bool
      retourne True sil y a au moins une sequence de deux elements consecutifs egaux,
      et False sinon.
      Precondition: x est une liste des entiers
      >>> sequenceDesDeux([1,4,3,3,4])
      True
      >>> sequenceDesDeux([1,2,3,4,3,2])
      False
     '''
     # ligne manque
          if x[i] == x[i+1]:
               return True
     return False
```

Quelle ligne doit-on y ajouter pour que la déscription soit correcte?

(a) `while i < len(x):`
(b) `while i < len(x) - 1:`
(c) `for i in range(len(x)):`
(d) `for i in range(len(x)-1):`
(e) `for i in range(len(x)-2):`

22. Le corps de la fonction suivante manque.

```
def meme_chaine(s1,s2):
     '''(str,str)->bool
     Precondition: len(s1)==len(s2) > 1
     Pour deux chaines de caracteres s1 et s2 de meme taille,
     la fonction retourne True si les deux chaines
     contiennent les memes caracteres et False sinon.
     '''
```

Quel code doit-on y ajouter pour que la déscription soit correcte?

(I)

```
for i in range(len(s1)):
    for j in range(len(s2)):
        if s1[i] != s2[j]:
            return False
return True
```

(II)

```
for i in range(len(s1)):
    if s1[i]==s1[i]:
        return True
return False
```

(III)

```
for i in range(len(s1)):
    if s1[i]!=s1[i]:
        return False
return True
```

(a) I seulement
(b) II seulement
(c) III seulement
(d) I et II
(e) I et III

23. Lesquels des programmes suivants I, II, et III affichent le même message que ce programme?

```
x = 0
if x < 20:
    print(x, "inferieur a 20")
else:
    if x > 20:
        print(x, "superieur a 20")
    else:
        print(x, " est 20")
```

I

```
x=0
if x < 20:
    print(x, "inferieur a 20")
else x > 20:
    print(x, "superieur a 20")
else:
    print(x, "est 20")
```

II

```
x=0
if x < 20:
    print(x, "inferieur a 20")
if x > 20:
    print(x, "superieur a 20")
else:
    print(x, " est 20")
```

III

```
x=0
if x < 20:
    print(x, "inferieur a 20")
elif x > 20:
    print(x, "superieur a 20")
else:
    print(x, " est 20")
```

(a) I seulement
(b) II seulement
(c) III seulement
(d) II et III
(e) I, II, et III

24. Laquelle est la déscription correcte de la fonction kiwi?

```
def kiwi(x):
     '''(list)->bool
        Precondition: les elements de la liste x sont des nombres
        et x a au moins 2 elements
     '''
     result=True
     for i in range(len(x)-1):
          if(x[i] > x[i+1]):
               result=False
          else:
               result=True
     return result
```

(a) La fonction retourne True si les nombres de la liste x sont en ordre ascendant, et False sinon.

(b) La fonction retourne False si l'avant-dernier élément est supérieur au dernier élément, et True sinon.

(c) Aucune des déclarations ci-dessus.

25. Combien d'étoiles sont affichées par le programme suivant?

```
n = 62
for number in range( n//3 ):
    print("*")
```

(a) 19 (b) 20 (c) 21 (d) 22 (e) 31

26. Combien d'étoiles sont affichées par le programme suivant?

```
for i in range(-1, 9):
     for j in range(10):
          print("*")
```

(a) 121 (b) 110 (c) 100 (d) 90 (e) 81

27. Combien d'étoiles sont affichées par le programme suivant?

```
x = 10
y = 100 % 90
i = 0
while i < 10000:
     if x!=y:
          print("*")
     i=i+1
```

(a) 0 (b) 1 (c) 10000 (d) 99999 (e) 10001

28. Laquelle est la déscription correcte de la fonction suivante?

```
def fig(lst):
     '''(list)->bool
       Precondition: les elements de lst sont des nombres'''
     if(len(lst)<2):
         return True
     d = 0
     for i in range(0, len(lst)-1):
        if lst[i+1] - lst[i] != d:
            return False
     return True
```

(a) Verifie si les nombres de la liste sont en ordre descendant

(b) Verifie si les nombres de la liste sont en ordre ascendant

(c) Verifie si les nombres de la liste sont tous égaux

(d) Verifie si les nombres de la liste sont tous differents

(e) Verifie si les nombres de la liste sont tous non-zero

29. La documentation de la fonction `cryp` explique ce qu'elle fait. Une ligne de code manque dans son corps, voir le commentaire `#ligne manque`. Quelle ligne doit-on y ajouter pour que la déscription soit correcte?

```
def cryp(s):
     '''(str)->str
      Retourne une nouvelle chaine de caracteres similaire a la chaine s
      mais les blocs de 3 caracteres consecutifs sont inverses.
      Si len(s) n'est pas multiple de 3, le dernier bloc (1 ou 2 caracteres) n'est pas retenu.

      >>> cryp("abc")
      'cba'
      >>> cryp("abc1")
      'cba'
      >>> cryp("abc12")
      'cba'
      >>> cryp("abc123def")
      'cba321fed'
      >>> cryp("123456789*$#")
      '321654987#$*'
     '''

     sc=''
     # ligne manque
          sc=sc+s[i+2]+s[i+1]+s[i]
     return sc
```

(a) `for i in range(len(s)):`

(b) `for i in range(3, len(s)):`

(c) `for i in range (0, len(s), 3):`

(d) `for i in range (0,len(s)-1, 3):`

(e) `for i in range(0,len(s)-2,3):`

30. Qu'est-ce que le programme Python suivant va afficher sur l'écran?

```
def f(x, B, C):
   x = 1
   B[0] = 2
   B=C
   B[0]= 3

x = 111;
Y = [10, 20]
Z = [30, 40]
f(x, Y, Z);
print(x, Y, Z)
```

(a) `111 [10, 20] [30,40]`
(b) `1 [10, 20] [30, 40]`
(c) `1 [3, 20] [3,30]`
(d) `111 [2, 20] [3, 40]`
(e) `111 [3, 20] [3,30]`

31. Une partie de la fonction suivante manque.

```
def common_chars(s1, s2):
    '''(str, str) -> str

    Retourne une nouvelle chaine avec les caracteres de s1 qui arrivent
    au moins une fois en s2. Les caracteres dans le resultat conservent
    leur ordre d'occurrence en s1.

    >>> common_chars('abc', 'ad')
    'a'
    >>> common_chars('a', 'a')
    'a'
    >>> common_chars('abb', 'ab')
    'abb'
    >>> common_chars('abracadabra', 'ra')
    'araaara'
    '''

    res = ''

    # code manque

    return res
```

Quel est le fragment de code qui manque?

(a)
```
for ch in s1:
    for ch in s2:
        res = res + ch
```

(b)
```
for ch in s2:
    if ch in s1:
        res = res + ch
```

(c)
```
for ch in s1:
        if ch in s2:
            res = res + ch
```

(d)
```
if ch in s2:
         for ch in s1:
             res = res + ch
```

32. Une partie de la fonction suivante manque.

```
def neg(l):
     '''(list of int)->list of int
     Retourne une liste avec tous les entiers x de l pour lesquels -x est aussi en l
     >>> neg([1, 1, 100, -1, -20, -300, 20, 55, 55])
     [1, 1, -1, -20, 20]
     >>> neg([-1, 100, 55, 55])
     []
     '''
     l2=[]

     # code manque

     return l2
```

Quel est le fragment de code qui manque?

I)

```
for x in l:
    if -x in l:
        l2.append(x)
```

II)

```
for x in l:
    if x<0 and x in l:
        l2.append(-x)
```

III)

```
for x in l:
    pas_ajoute=True
    for y in l:
        if x+y==0 and pas_ajoute:
            l2.append(x)
            pas_ajoute=False
```

(a) I et II
(b) I et III
(c) I seulement
(d) II seulement
(e) III seulement

33. Laquelle est la déscription correcte de la fonction suivante?

```
def ma_fonction(n):
     ''' (int)->(int)
     Precondition: n>=1'''
     for i in range(n):
          total=1
          total=total*n
     return total
```

(a) elle retourne n^n

(b) elle retourne $n * n$

(c) elle retourne n

(d) elle retourne 1

(e) elle donne une erreur pour $n \geq 1$