DS de simulation numérique avec Python (B)

1h, sans document, sans calculatrice ou autre système électronique

A. Ammar

Nom : _____ **Prénom :** _____ **Date :**_____

Exercice N°1 (6 pts) :

1) Pour chacune des variables **var1** à **var5**, indiquer son type Python.

Variable	Type	
`var1 = "23"`	`str`	(0,5)
`var2 = 3`	`int`	(0,5)
`var3 = 1.0`	`float`	(0,5)
`var4 = [1, 2, 3.14, 2018]`	`list`	(0,5)
`var5 = var2/var3`	`float`	(0,5)

2,5

2) Soit une liste **maliste** définie de la façon suivante: **`maliste = [True, "y", 3.14, 2+3, 18]`**

Pour chacune des variables **var1** à **var5** , indiquer sa valeur :

Variable	Valeur	
`var1 = maliste[-1]`	`18`	(0,5)
`var2 = maliste[1:3]`	`['y', 3.14]`	(0,5)
`var3 = maliste[0:4]`	`[True, 'y', 3.14, 5]`	(0,5)
`var4 = maliste[:3]`	`[True, 'y', 3.14]`	(0,5)
`var5= maliste[-3:-1]`	`[3.14, 5]`	(0,5)

2,5

3) Soit une variable **age** contenant une valeur numérique entière. Utiliser l'appel à la fonction **`print()`** afin d'afficher la valeur de la variable **age** sous la forme (exemple avec **`age = 25`**) :
`Vous avez 25 ans`

```
print("Vous avez", age, "ans")
# ou bien                                          (1)
print("Vous avez "+str(age)+" ans")
# ou encore
print("Vous avez {} ans".format(age))
```

1

Exercice N°2 (4 pts) :

Écrire un programme qui permette de saisir un nombre réel; s'il est positif ou nul alors afficher sa racine carrée, sinon afficher un message d'erreur. On considère que l'utilisateur saisit un nombre réel.

```
from math import sqrt        (1)
a = float(input("Donnez un nombre réel : "))  (1)
if a >= 0 :
    print("La racine carrée de {} = {}".format(a, sqrt(a))) (1)
else:
    print("Erreur! Entrez un nombre positif.")  (1)
```

4

Exercice N°3 (4 pts) :

1) Calculer la valeur de π en utilisant une boucle **while** pour la formule de *Leibniz* suivante :

$$\frac{\pi}{4} = \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1}$$

2) Comparer la valeur trouvée de π avec celle de la bibliothèque **math**.

```
n = 10000          (0,5)
my_pi = 0   # initialisation  (0,5)
k = 0  # initialisation du compteur k  (0,5)
while k <= n:
    my_pi += (-1)**k/(2*k+1.0) (0,5)
    k += 1   (0,5)
my_pi *= 4    (0,5)
print(my_pi)
```

3

```
from math import pi   (0,5)
print(abs(pi - my_pi))   (0,5)
```

1

Exercice N°4 (6 pts) :

1) Définir une séquence de nombres: $x_n = (n-1)^2 + 1$, pour les entiers $n = 1,2,\ldots, N$. Écrivez un programme qui affiche x_n pour $n = 1,2,\ldots, 10$ en utilisant une boucle **for**.

```
for n in range(1, 11) :  (1)
    x_n = (n - 1)**2 + 1   (0,5)
    print('x{} = {}'.format(n, x_n))  (0,5)
```

2

2) Stocker **uniquement les valeurs impaires** calculées de x_n dans une liste (à l'aide d'une boucle **for**).

```
x = [] (0,5)
for n in range(1, 11) :    (1)
    x_n = (n - 1)**2 + 1   (0,5)
    if x_n % 2 != 0:       (0,5)
        x.append(x_n)       (0,5)
```

3

3) Écrire une fonction Python **x(n)** pour calculer un élément dans la séquence $x_n = (n-1)^2 + 1$. Appeler la fonction pour n = 2 et afficher le résultat.

```
def x(n):
    return (n - 1)**2 + 1   (0,5)
print(x(2))     (0,5)
```

1