

Algorithmique et programmation 1

CT - Mardi 4/01/2022 de 9h45 à 11h45

- Aucun document n'est autorisé, ni aucun matériel électronique (ordinateurs, calculatrices et/ou téléphones portables).
- L'examen est noté sur 20 points, le barème est donné à titre indicatif.
- Les programmes doivent être écrits en Python 3, en utilisant uniquement les notions vues en cours. Par conséquent, l'utilisation des fonctions, ... ou tout autre spécificité du langage est interdite.
- Les **réponses 2.1, 3 et 5.1** doivent contenir une entête incluant le rôle et la liste des variables (nom et type).

Exercice 1. Questions de cours

1 + 1 + 1 + 0.5 = 3.5 pts

- 1. Donner une définition de la notion de *variable*, et une de celle d'*affectation*.
- 2. J'ai perdu mon support de cours, et je ne me souviens plus de la définition complète d'un tableau 1D : *Un tableau 1D est une ... (1) ... unidimensionnelle qui permet de stocker un ensemble de valeurs de ... (2) ... en leur associant un ... (3) L'accès à chacune de ces valeurs se fait par un ... (4) Compléter cette définition.*
- 3. Quel est le résultat de l'exécution du programme Python suivant, pour $a = 17$?

```
# Variables
#   a, b, c : entier
#   d : booléen
a = int(input("Valeur de a= ")) # valeur lue : 17
b = ((a // 5) - 1) * 5 + 5
c = ((a / 5) - 1) * 5 + 5
d = (b == c)
print("Finalement b = ", b, ", c = ", c, " et d = ", d, ".")
```

- 4. Le résultat de l'exécution du programme précédent aurait-il été différent si la variable a avait été lue comme un réel? Pourquoi.

Exercice 2. Carte de photocopies

1 + 0.5 = 1.5 pts

Pour faire des photocopies à la BU, vous pouvez utiliser une carte pré-payée, contenant un certain nombre d'unités, une unité donnant droit à une photocopie A4/recto. Pour pouvoir faire des photocopies, vous devez donc recharger votre carte, au prix de 0.2 euro l'unité.

- 1. Écrire un programme qui lit le nombre d'unités que vous souhaitez recharger sur la carte, puis qui calcule et affiche le prix en euros de la recharge.

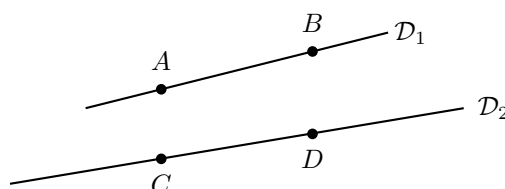
Depuis le 3/01/2022, il n'est plus possible de recharger que 100, 200, 500 ou 1000 unités.

- 2. Que faut-il ajouter au programme précédent pour n'afficher le prix que si le nombre d'unités souhaitées est valide (100, 200, 500 ou 1000), et l'indiquer à l'utilisateur si ce n'est pas le cas.

Exercice 3. Droites parallèles, perpendiculaires ou quelconques?

2 pts

Soient une droite \mathcal{D}_1 passant par les points $A(x_A, y_A)$ et $B(x_B, y_B)$, et une droite \mathcal{D}_2 passant par les points $C(x_C, y_C)$ et $D(x_D, y_D)$, comme illustré ci-dessous.



Leurs coefficients directeurs sont notés c_1 et c_2 , respectivement, avec par exemple $c_1 = (y_B - y_A)/(x_B - x_A)$.
Finalement les deux droites sont :

- parallèles : si elles ont le même coefficient directeur, c'est-à-dire, $c_1 = c_2$,
- perpendiculaires : si elles ont des coefficients directeurs qui sont l'opposés de l'inverse l'un de l'autre, c'est-à-dire, $c_1 = -1/c_2$,
- et quelconques sinon.

- 1. Écrire un programme qui lit les coordonnées des points A, B, C et D , puis qui détermine et affiche si les droites \mathcal{D}_1 et \mathcal{D}_2 sont parallèles, perpendiculaires ou quelconques.

Exercice 4. Structures conditionnelles imbriquées

1 + 1 = 2 pts

Soit le programme Python suivant.

```

1  # Variables
2  #   x, r : entier
3  x = int(input("x= "))
4  r = (x + 17) // 34
5  if r < 1:
6      print("A", end="")
7      if r != 0:
8          print("B")
9      else:
10         print("C")
11 else:
12     print("D", end="")
13     if r != 0:
14         print("E")
15     else:
16         print("F")

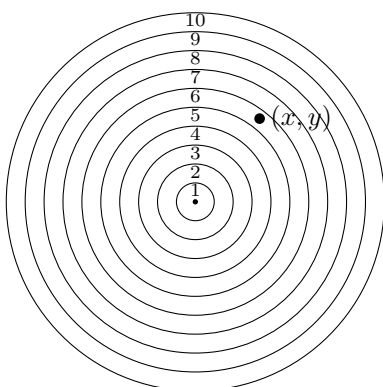
```

- 1. Quelle condition doit vérifier x pour que le test de la ligne 5 soit vérifié.
- 2. Donner 4 valeurs x pour lesquels le programme Python affichera : AB, AC, DE, et DF.
Si ce n'est pas possible, proposer une explication.

Exercice 5. En plein dans le 1000

2 + 1 = 3 pts

Une cible de centre de coordonnées $(0, 0)$ est constituée de 10 cercles concentriques, de rayons variant de 1 à 10. Une flèche est lancée sur la cible aux coordonnées (x, y) , dans le cercle de rayon r , comme illustré sur le schéma ci-dessous où $r = 6$.



En fonction du cercle dans lequel la flèche atterrit, elle rapporte un certain nombre de points n défini de la manière suivante :

$$n = \begin{cases} 50 & \text{si } (x, y) = (0, 0), \\ 20 - r & \text{si } r \in \{1, \dots, 10\}, \\ 0 & \text{sinon.} \end{cases}$$

Remarque : si la flèche atterrit exactement sur le cercle de rayon r , elle est considérée à l'intérieur du cercle de rayon r , et rapporte également $n = 20 - r$ points.

- 1. Écrire un programme qui lit les coordonnées (x, y) de la flèche, puis détermine de manière itérative le rayon r du cercle dans lequel elle atterrit, et affiche le nombre de points qu'elle rapporte.
- 2. Modifier le programme précédent pour effectuer le même traitement sans utiliser de structure itérative.

Exercice 6. Une somme d'entiers un peu particulière

1 + 1 = 2 pts

Soit $n > 0$ un entier lu au clavier. On souhaite calculer la somme suivante de manière itérative :

$$S = 1^1 + 2^2 + \dots + n^n.$$

- 1. Écrire un programme qui lit la valeur n , puis qui calcule et affiche la somme S ci-dessus, en utilisant des boucles tant-que, et sans utiliser l'opérateur puissance du langage Python.
- 2. Est-il possible d'utiliser des boucles pour? Si oui, modifier le programme précédent. Si non, proposer une explication.

Exercice 7. Programme simple sur les tableaux

2 pts

Soit t un tableau 1D de taille n , définit de la manière suivante :

$$\forall i \in \{1, \dots, n\}, t[i] = i \times t[i-1], \text{ avec } t[0] = 1.$$

- 1. Écrire un programme qui lit la taille n du tableau, puis qui construit et affiche le tableau t .

Exercice 8. Permutation

2 pts

Soit t un tableau 1D de taille n . Le tableau t est une permutation de taille n si

$$\forall e \in \{0, \dots, n-1\}, \exists i \in \{0, \dots, n-1\} \mid t[i] = e.$$

- 1. Écrire un programme qui lit la taille n du tableau, et le tableau t de taille n , puis qui détermine et affiche si le tableau t est une permutation de taille n .

Exercice 9. Un peu d'algèbre linéaire

2 pts

Soit M une matrice de taille $n \times m$, représentée par un tableau 2D. La transposée de M , notée M^T , est une matrice de taille $m \times n$ définie de la manière suivante :

$$\forall (i, j) \in \{1, \dots, n\} \times \{1, \dots, m\}, M_{i,j} = M_{j,i}^T.$$

- 1. Écrire un programme qui lit les valeurs n, m , qui construit et affiche la matrice M , puis qui construit et affiche la transposée M^T .