

Examen Médian

Durée : 1 heure 30

Feuille A4 recto-verso autorisée, autres documents interdits.
Calculatrices, téléphones, traducteurs et ordinateurs interdits.

Attention : chaque partie doit être rédigée sur une copie séparée

1ère partie : choix

1.1. Quelle salle pour le médian ?

Ecrire un programme en Python qui demande à un étudiant de INF1 la première lettre de son nom de famille et qui lui indique la salle dans laquelle a lieu le médian. On supposera que les étudiants dont la première lettre du nom est comprise entre 'A' et 'H' doivent aller dans la salle FA501, ceux dont la première lettre est comprise entre 'I' et 'P' dans la salle FA502 et les autres dans la salle FA503.

```
lettre = input('Première lettre de votre nom ? ')
lettre = lettre.upper()

while (lettre < 'A') or (lettre > 'Z'):
    lettre = input('Saisie incorrecte\nPremière lettre de votre
nom ? ')
    lettre = lettre.upper()

if lettre <= 'H' :
    salle = 'FA501'
elif lettre <= 'P':
    salle = 'FA502'
elif lettre <= 'Z':
    salle = 'FA503'

print('Numéro de salle :', salle)
```

1.2. Un programme mystérieux

On considère le programme Python suivant :

```
print('Saisir deux entier compris entre 0 et 20')
boule = int(input())
de = int(input())
gomme = boule
if boule < 10 :
    gomme = -boule
print(gomme)
if de < 5 :
    gomme = gomme - de
print('Résultat', gomme)
```

- a) Quel est le résultat de ce programme si l'utilisateur saisit 2 et 3 ? -5
b) Quel est le résultat de ce programme si l'utilisateur saisit 15 et 2 ? 13

1.3. Le jeu de Chifoumi

Le jeu de Chifoumi, aussi appelé caillou-ciseaux-papier, se joue à deux joueurs avec les mains. Simultanément, les deux joueurs font un signe avec leur main qui représente soit un caillou, soit des ciseaux, soit un papier. Si on nomme les joueurs A et B, les règles sont les suivantes :

- Si A et B font le même signe, il y a égalité, aucun des deux joueurs ne marque de point.
- Si le joueur A joue Caillou et le joueur B Ciseaux, A marque un point, car « le caillou émousse les ciseaux », et réciproquement.
- Si A joue Papier et B joue Caillou, A marque un point, car « le papier enveloppe le caillou », et réciproquement.
- Si A joue Ciseaux et B joue Papier, A marque un point car « les ciseaux coupent le papier », et réciproquement.

- 1) Ecrire un programme Python qui permet à l'utilisateur de saisir les deux coups joués par A et B et qui affiche le nom du joueur qui marque un point.)

```
# Caillou - ciseaux - papier v1

# lecture joueurA
lu = False

while not lu :
    jeuA = input('Caillou (1), Ciseaux (2), Papier (3) ')
    lu = jeuA in {'1', '2', '3'}

# lecture joueurB
lu = False

while not lu :
    jeuB = input('Caillou (1), Ciseaux (2), Papier (3) ')
    lu = jeuB in {'1', '2', '3'}

if jeuA == jeuB:
    print('égalité')
elif (jeuA == '1' and jeuB == '2') or (jeuA == '2' and jeuB == '3')
or (jeuA == '3' and jeuB == '1'):
    print('A gagne')
else :
    print('B gagne')
```

- 2) Modifier le programme précédent pour ajouter un score à A et B, et arrêter la partie quand l'un des joueurs atteint 5 en indiquant qui est le gagnant.

```
# Caillou - ciseaux - papier avec points

scoreA = 0
scoreB = 0
fini = False
```

```

while not fini:
    # lecture joueurA
    lu = False

    while not lu :
        jeuA = input('Joueur A : Caillou (1), Ciseaux (2), Papier (3)
        ')
        lu = jeuA in {'1', '2' , '3'}

    # lecture joueurB
    lu = False

    while not lu :
        jeuB = input('Joueur B : Caillou (1), Ciseaux (2), Papier (3)
        ')
        lu = jeuB in {'1', '2' , '3'}

    if jeuA == jeuB:
        print('égalité')
    elif (jeuA == '1' and jeuB == '2') or (jeuA == '2' and jeuB ==
    '3') or (jeuA == '3' and jeuB == '1'):
        print('A gagne')
        scoreA = scoreA + 1
    else :
        print('B gagne')
        scoreB = scoreB + 1

    fini = scoreA == 5 or scoreB == 5

if scoreA == 5:
    print('Vainqueur A :', scoreA, 'à', scoreB)
else:
    print('Vainqueur B :', scoreB, 'à', scoreA)

```

----- { *prendre une nouvelle copie* } -----

2^{ème} partie : sapin numérique

On souhaite réaliser un programme en Python qui donne la possibilité à l'utilisateur d'afficher à l'écran une figure 'sapin numérique' (voir Figure 1 ci-dessous).

```

      1
     1 1
      2
    1 2 1
     2 2
      3
    1 2 3 1
     2 3 2
      3 3
     4
    1 2 3 4 2 1
     2

```

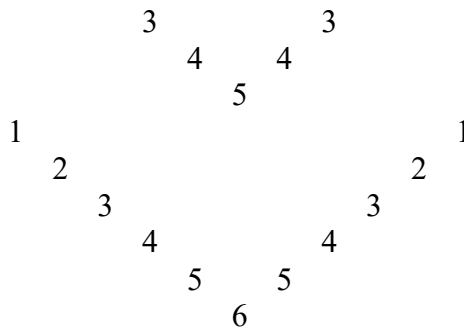


Figure 1. Sapin numérique

L'utilisateur doit pouvoir définir la taille du sapin (le nombre d'étages est 6 pour la Figure 1).

Pour écrire le programme, on procédera en 3 étapes :

1. Définir le nombre n ($1 < n < 10$) d'étages du sapin. L'utilisateur peut se tromper dans sa demande (n en dehors du domaine de définition), mais pas plus de 5 fois.
2. Ecrire un programme Python permettant d'afficher à l'écran un étage de branche. Par exemple pour l'étage 4, le programme affichera :

```

      1           1
     2         2
    3       3
   4

```

3. Afficher le sapin avec le nombre d'étages demandé par l'utilisateur en utilisant l'étape 2.

```

# sapin numérique
def nbEtages():
    "retourne un nb d'étages strictement entre 1 et 10"

    nb = 0
    i = 0    # nb de répétitions limité à 5

    while (nb <= 1 or nb >= 10) and (i < 5):
        nb = int(input("Combien d'étages ? "))
        i = i + 1
    if 1 < nb < 10 :
        return nb
    else:
        return 0

def etage(n,marge):
    "affiche l'étage n avec une marge de marge caractères"
    for i in range (1,n):
        print(" "*(i -1 + marge), i, " "*(2 * (n - i) - 1), i)
    print(" "*(n+marge), n)

# corps principal
nb = nbEtages()
if nb :

```

```
for i in range(1, nb+1):
    etage(i, nb - i)
```

----- { prendre une nouvelle copie } -----

3^{ème} partie : Tableaux

- a) Ecrire un algorithme permettant à l'utilisateur de saisir un tableau de 8 caractères, puis réalisant un décalage circulaire vers la gauche des éléments de ce tableau.

Exemple :

Tableau initial : D E C A L A G E

Tableau modifié (décalage à gauche) : E C A L A G E D

constante N = 8

variables tmp : caractere

i : entier

T : tableau de N caractères

Début

Pour i allant de 0 à N-1 faire

 Lire T[i]

Fin pour

tmp <- T[0]

Pour i allant de 0 à N-2 Faire

 T[i] <- T[i+1]

Fin pour

T[N-1] <- tmp

Fin

Ou (si on considère que le premier indice est 1)

constante N = 8

variables tmp : caractere

i : entier

T : tableau de N caractères

Début

Pour i allant de 1 à N faire

 Lire T[i]

Fin pour

tmp <- T[1]

Pour i allant de 1 à N-1 Faire

 T[i] <- T[i+1]

Fin pour

T[N] <- tmp

Fin

b) Ecrire un algorithme permettant à l'utilisateur de tester si un tableau est trié en ordre croissant.

constante N=10

variables i : entier

T : tableau de N entiers

Début

i <- 0

Tant que i < N-1 ET T[i] <= T[i+1] Faire

i <- i + 1

Ftq

Si i = N-1 alors

écrire 'Le tableau est trié'

Sinon

écrire 'Le tableau est non trié'

Fsi

Fin

ou

constante N=10

variables i : entier

T : tableau de N entiers

Début

i <- 1

Tant que i < N ET T[i] <= T[i+1] Faire

i <- i + 1

Ftq

Si i = N alors

écrire 'Le tableau est trié'

Sinon

écrire 'Le tableau est non trié'

Fsi

Fin