

Série d'exercices 01**Algorithmique et Programmation**4^{ème} SI 1

Zied Fridhi

Exercice 01 :

Dans un contexte informatique et pour chacune des propositions suivantes mettre V si correcte et F sinon :

1) Quel est le type de la variable a ? (a=33.0) :

☐

int

☐

float

☐

str

☐

bool

2) Qu'affiche le programme suivant ?

```
a = 12 * 4.0
print (a)
```

☐

12 * 4.0

☐

48

☐

48.0

☐

Erreur

3) Qu'affiche le programme suivant ?

```
test = 1 + 1 / 2
print ("test")
```

☐

1.5

☐

1,5

☐

3 / 2

☐

test

4) Cocher les nombres qu'ont peut obtenir en exécutant cette séquence :

```
from random import *
n = randint (10,19)
print (n)
```

☐

3

☐

19

☐

10

☐

21

☐

13

☐

17

5) Que contient la variable (a) sachant que :

```
ch ← "bonsoir"
a ← pos ("a", ch)
```

☐

0

☐

-1

☐

"a"

☐

Erreur

6) Qu'affiche le programme suivant :

```
ch = "test"
a = ch.find ("t")
print (a)
```

☐

0

☐

1

☐

3

☐

0,3

7) Qu'affiche l'algorithme suivant (les "/" remplacent ici un retour à la ligne) :

```
Pour k de 0 à 3 faire
    Ecrire (k * k + k)
Fin pour
```

☐

2 / 6 / 12 / 20

☐

0 / 1 / 4 / 9

☐

0 / 1 / 2 / 3

☐

0 / 2 / 6 / 12

8) Quelle est la valeur de la variable n à la fin de l'exécution du programme suivant :

```
n = 1
for i in range (4) :
    n += 2
```

☐

1

☐

8

☐

9

☐

18

9) Qu'affiche l'algorithme suivant :

```
a ← 0
```

```
Pour i de 5 à 9 faire
```

```
  a ← a + 1
```

```
Fin pour
```

```
Ecrire (a)
```

 1 4 5 6

10) Que contient la variable x sachant que :

```
x ← 0
```

```
Pour i de 0 à 9 faire
```

```
  Pour j de -1 à -9 (pas = -1) faire
```

```
    x ← x + 1
```

```
  Fin pour
```

```
Fin pour
```

 90 99 100 109

11) Que contient la variable (ch2) sachant que :

```
ch1 ← "PYTHON"
```

```
ch2 ← majus (ch1)
```

 "ch1" "python" "Python" "PYTHON"

12) Qu'affiche le programme suivant :

```
ch1 = "TesT"
```

```
ch2 = ch1.upper ()
```

```
print (ch2)
```

 "Test" "TesT" "TEST" "TeST"

13) On considère l'extrait de code suivant :

```
while (a < 20) or (b > 50) :
```

```
  .....
```

```
  .....
```

Quelles conditions permettent de mettre fin à cette boucle ?

☐ la boucle prend fin lorsque a < 20 ou b > 50☐ la boucle prend fin lorsque a < 20 et b > 50☐ la boucle prend fin lorsque a ≥ 20 ou b ≤ 50☐ la boucle prend fin lorsque a ≥ 20 et b ≤ 50

14) Que contient la variable n sachant que :

```
n ← 1
```

```
Tant que (n ≠ 20) faire
```

```
  n ← n + 2
```

```
Fin tant que
```

☐ 1☐ 20☐ 22☐ le programme ne termine pas, la boucle tourne indéfiniment

Exercice 02 :

Compléter l'algorithme du module suivant :

Procédure Quoi (@ f1, f2 :texte)

Début

..... ("d:\travail\texte1.txt",,) # ouvrir f1 en mode lecture

..... ("d:\travail\texte2.txt",,) # ouvrir f2 en mode écriture

Tant que Non(fin_fichier(f1)) faire

 Lire_ligne (f1,ch)

 i ← 1

 Tant que (i ≤ long(ch)) faire

 aux ← ch[i-1]

 ch[i-1] ← ch[i]

 ch[i] ← aux

 i ← i+2

 Fin Tant que

 Ecrire_nl (f2, ch)

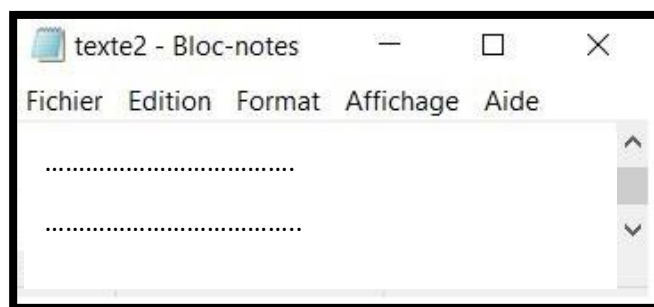
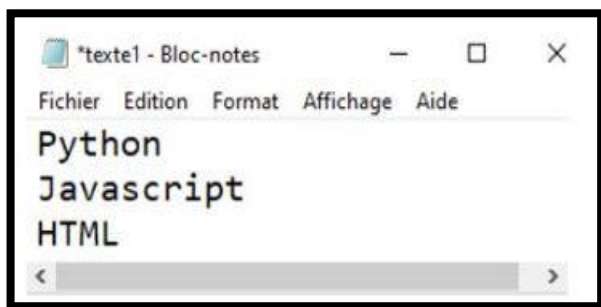
Fin Tant que

.....

.....

Fin

- a- Donner le contenu du fichier f2 sachant que le fichier f1 contient les 3 lignes suivantes :
- | | |
|------------|------------|
| texte1.txt | texte2.txt |
|------------|------------|



- b- Proposer un rôle pour cette procédure.

.....

- c- On veut sauvegarder dans un fichier d'enregistrements, chaque ligne du fichier texte1.txt avec une description pour chaque langage. Proposer une déclaration d'objets (tnt/tdo)

Exercice 03 :

Écrire un programme qui permet de calculer et d'afficher la somme des factorielles des chiffres d'un entier **n** donné ($10 \leq n \leq 50$).

Exemples :

Saisir un entier : 27 le programme affichera : $2! + 7! = 5042$

Saisir un entier : 39 le programme affichera : $3! + 9! = 362886$

Exercice 04 :

Un entier naturel écrit dans la base 10 est dit rond si, dans son équivalent binaire, on trouve autant de 1 que de 0.

Exemple: le nombre 611 est rond car $611 = (1001100011)_2$ et il y a autant de 1 que de 0 dans cette écriture en binaire.

Écrire un programme qui permet de chercher et d'afficher tous les nombres ronds de l'intervalle $[10, 1000]$.

Exercice 05 :

On se propose d'afficher les n premières lignes du triangle de pascal **M**.

Le triangle de pascal est un tableau triangulaire construit de la manière suivante :

- Les éléments de la première colonne sont tous égaux 1.
- Les éléments de la diagonale (**ligne = colonne**) sont tous aussi égaux à 1.
- Pour les autres éléments :

$$M[\text{ligne}, \text{colonne}] = M[\text{ligne}-1, \text{colonne}-1] + M[\text{ligne}-1, \text{colonne}]$$

Exemple :

Pour $n = 5$:

	0	1	2	3	4
0	1				
1	1	1			
2	1	2	1		
3	1	3	3	1	
4	1	4	6	4	1

$$M[4,1] = M[3,0] + M[3,1]$$

Écrire l'algorithme d'un module qui permet de remplir les n premières lignes du triangle de pascal.