Série de Rappel N°2

Objectifs à atteindre :

Se rappeler :
1. Des structures itératives Complète.
2. Des structures itératives à condition d'arrêt.

RAPPEL

		Algorithme	Python
Structure iterative complete	Pour	**Pour** compteur **de** debut **à** fin **faire** Traitement **FinPour**	**for** compteur **in range**(debut,fin): Traitement
Structure iterative à condition d'arrêt	Répéter	**Répéter** Traitement **Jusqu'à (condition)**	**while** not(Condition) : Traitement
	TantQue	**TantQue** (Condition) **Faire** Traitement **Fin TantQue**	**while** condition : Traitement

Exercice N° 1

Déterminer manuellement les affichages générés par chacune des séquences A, B, C, D, E et F.

N°	Séquence	Affichage
A	`for i in range(1,3):` `  print(i)`	

B	`for i in range(1,3):`
	`  print(i)`
	`print(i)`	

C	`for i in range(1,3):`
	`  print(i)`
	`  print(i)`	

D	`for i in range(3,1):` `  print(i)`	
E	`for i in range(3,1,-1):` `  print(i)`	
F	`for i in range(1,3):` `  print(i,end=',')`	

Exercice N° 2

1. Donner la valeur de i donnée par les deux algorithmes suivants :

Algo alog_repeter debut i←0 répéter i←i+1 Jusqu'à i=5 fin	Algo algo_tant_que Debut i←0 Tant que (i=5) faire i←i+1 Fin tant que fin	
i=	i=	***Algorithme modifié***

2. Si les deux résultats (i) ne sont pas identiques, modifier un de ces algorithmes pour avoir le même résultat.

Exercice N° 3

Exécuter manuellement ces instructions.

x←8 **répéter** x←x+2 y←x*2 **jusqu'à** y>25	p←0 **Tant que** p <5 **faire** p←p+2 **Fin Tant que**	p←0 **Tant que** p> 5 **faire** p←p+5 **Fin Tant que**	x←5 **répéter** x←x+2 k←x*2 **jusqu'à** k <30
x= y=	p=	p=	x= k=
Nombre d'itérations :	Nombre d'itérations :	Nombre d'itérations :	Nombre d'itérations :

Exercice N° 4

Dans un contexte informatique et pour chacune des propositions données ci-dessous, mettre dans chaque case la lettre **V** si la proposition est correcte ou la lettre **F** dans le cas contraire.

1. Une chaine « ch » est dite « joli-mot » si le nombre de ses voyelles est égale au nombre de ses consonnes.
   Exemple : ch ←"Joud" :Nombre des voyelles=Nombre des consonnes.
   Afin de vérifier si ch déjà saisie est un joli-mot ou non, on utilise ce bloc d'instructions :

☐
```
nv←0, nc←0
Pour i de 0 à long (ch) -1 faire
  Si Majus (ch[i]) dans ["A","E","Y","U","I","O"] Alors
      nv← nv+1
  Sinon nc← nc+1
  FinSi
Fin Pour
Si nc=nv alors
    écrire ("Joli-Mot")
Sinon
écrire ("Non")
FinSi
```

☐
```
nv←0, nc←0
Pour i de 0 à long (ch) -1  faire
Si Majus (ch[i]) dans ["A"..."Z"] Alors
 Si Majus (ch[i]) dans ["A","E","Y","U","I","O"] Alors
   nv ← nv+1
 Sinon
    nc←  nc+1
 FinSi
 FinSi
 Fin Pour
Si nc=nv alors
   écrire ("Joli-Mot")
Sinon écrire ("Non")
FinSi
```

2. Une chaine est dite « **Pangramme** » si elle contient toutes les lettres de l'alphabet sans distinction entre majuscule et minuscule.
   Exemple : ch←"**Dans un wagon bleu, tout en mangeant cinq kiwis frais, vous jouez du xylophone**"

11

a) Pour vérifier si une chaine donnée est pangramme ou non, quelle est la boucle (plus optimale) à utiliser.

☐ La boucle POUR ☐ La boucle REPETER ☐ La boucle TANT QUE

b) Pour vérifier si une chaine « ch » donnée est **une pangramme** ou non on utilise ce fragment du programme suivant :

☐
```
ok← vrai
Pour i de "a" à "z" faire
  Si Non ((Pos (i, ch) <>-1) ou (Pos (Majus (i), ch))) < >-1)
    Alors ok ← faux
  FinSi
  FinPour
Si ok alors
        ecrire ("Pangramme")
Sinon ecrire ("Non")
FinSi
```

☐
```
ch1←""
Pour i de 0 à long (ch) -1 Faire
C← Majus (ch[i])
   si (c dans ["A".."Z"]) et (Pos (C , ch1) =-1)
     Alors ch1← ch1+C
  FinSi
FinPour
Si long ( ch1 )=26 alors
        ecrire ("Pangramme")
Sinon ecrire ("Non")
FinSi
```

3. Un entier est dit « **Pur** » si le produit de ses diviseurs propres égales à lui-même.
   Exemple : N ← 6 : N est **Pur** car 1*2*3 =6
   Le bloc d'instructions qui permet de vérifier si un entier N donnée est **Pur** ou non :

☐
```
P←1
Pour i de 1 à n faire
  Si n MOD i =0 Alors
  P←P*i
  FinSi
FinPour
Si P=n alors
   ecrire ("Pur")
Sinon ecrire  ("Non")
FinSi
```

☐
```
P←1
Pour i de 1 à n div 2 faire
  Si n MOD i =0 Alors
  P←P*i
  FinSi
FinPour
Si P=n alors
ecrire ("Pur")
Sinon ecrire  ("Non")
Fin Si
```

☐
```
P←1
Pour i de 2 à n - 1 faire
  Si n MOD i =o Alors
  P ← P*i
  FinSi
FinPour
Si P=n alors
ecrire (''Pur'')
Sinon ecrire  (''Non'')
FinSi
```