TP2 - Boucle while

I Introduction

On utilise une boucle `while` lorsqu'on ne connaît pas le nombre de répétitions.
La syntaxe est :

```
while <condition> :
  <code>
```

La condition écrite après `while` est une expression booléenne qui vaut `True` ou `False`.

Exemple :

```
1 u = 1
2 while u < 5 :
3   u = 2 * u
4 print(u)
```

Commentaires :
On exécute la ligne 3 tant que la valeur de `u` est inférieure à 5.

II Expressions booléennes

On donne ci-dessous quelques exemples d'expressions booléennes avec `a` de type `int`.

- `a == 0` : vaut `True` si `a` est égal à 0 et `False` sinon.
- `a != 0` : vaut `True` si `a` n'est pas égal à 0 et `False` sinon.
- `a > 0 and a != 5` : vaut `True` si `a` est un entier strictement positif différent de 5 et `False` sinon.
- `a % 3 == 0 or a % 5 == 0` : vaut `True` si `a` est un multiple de 3 ou de 5 et `False` sinon.
- `not (a >= 0)` : vaut `True` si `a` est strictement négatif et `False` sinon.

III Exécution pas à pas

On considère le programme suivant :

```
1 n = 0
2 u = 1
3 while u < 5 :
4   u = 2 * u
5   n = n + 1
6 print(n)
```

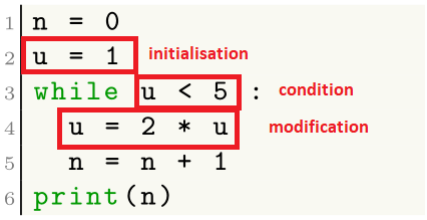
Q1. Compléter le tableau en exécutant pas à pas le programme ci-dessus.

ligne	1	2	3										
`n`	0												
`u`		1											
`u < 5`			`True`										
affichage													

IV Ecriture d'une boucle `while`

Remarque : Si on n'y prend pas garde, une boucle `while` peut être infinie (elle ne s'arrête pas).
Donc il y a 3 points essentiels à réfléchir lorsqu'on écrit une boucle `while` :

1. **l'initialisation** (de la variable intervenant dans l'expression booléenne) ;
2. **la condition** (c'est l'expression booléenne) ;
3. **la modification** (de la variable intervenant dans l'expression booléenne).



Comment écrire la condition :
On réfléchit à la condition en Français puis on la traduit en `Python`

- soit en disant : *on continue tant que ...*
- soit en disant : *on s'arrête quand ..., donc on continue tant que ...*

Q2. Ecrire un programme qui traduit la situation suivante :
On lance un dé jusqu'à ce qu'on tombe sur 6. A chaque lancer, on affiche le dé.
On s'arrête quand ....., donc on continue tant que .....

Q3. Ecrire un programme qui traduit la situation suivante :
On lance deux dés jusqu'à ce qu'on tombe sur une paire. A chaque lancer, on affiche les deux dés. Puis on indique le nombre de lancers qu'il a fallu pour obtenir la paire.
On s'arrête quand ....., donc on continue tant que .....

Q4. Ecrire un programme qui traduit la situation suivante :
Un enseignant doit rentrer une note comprise entre 0 et 20. S'il se trompe le programme indique un message d'alerte et lui redemande de rentrer la note.

Q5. Ecrire un programme qui traduit la situation suivante :
Un usager retire de l'argent à un distributeur de billets. Son code de carte bleue est 1234. Il n'a droit qu'à 3 essais.

Q6. Le juste prix
L'utilisateur doit deviner le nombre entier choisi aléatoirement entre 0 et 100 par l'ordinateur. L'ordinateur indiquera à chaque proposition de l'utilisateur « C'est plus » ou « C'est moins ».

Concaténation de 2 chaînes de caractères :
On peut concaténer (c'est-à-dire mettre bout à bout) deux chaînes de cacractères en utilisant l'opérateur `+`.

Exemple :

```
a = "Bonjour"
b = " !!!"
c = a + b #on concatène a et b
d = c + b
# c a pour valeur la chaîne de caractères "Bonjour !!!"
# d a pour valeur la chaîne de caractères "Bonjour !!! !!!"
```

Q7. Pour obtenir l'écriture binaire d'un entier, on effectue des divisions successives par 2.
Analyser l'exemple et écrire un programme qui demande à l'utilisateur de rentrer un entier et qui affiche l'écriture binaire de l'entier.
L'écriture binaire sera stockée dans une variable `b` de type `str`.

19	2	
1	9	2
	1	4	2
		0	2	2
			0	1	2
				1	0	STOP

L'écriture binaire de 19 est 10011.

Q8. Décomposition en nombres premiers.
Ecrire un programme qui demande à l'utilisateur de rentrer un entier supérieur ou égal à 2 et qui affiche la décomposition de l'entier en nombres premiers. Pour cela, analyser l'exemple (naïf) suivant :
La décomposition de 60 en nombres premiers est $2 \times 2 \times 3 \times 5$

- On regarde si 60 est divisible par 2. Oui ! Donc on le divise par 2 et on obtient le quotient 30 ;
- On regarde si 30 est divisible par 2. Oui ! Donc on le divise par 2 et on obtient le quotient 15 ;
- On regarde si 15 est divisible par 2. Non ! (A ce stade, les quotients suivants ne peuvent plus être divisibles par un multiple de 2)
- On regarde si 15 est divisible par 3. Oui ! Donc on le divise par 3 et on obtient le quotient 5 ;
- On regarde si 5 est divisible par 3. Non !
- On regarde si 5 est divisible par 4. Non ! (Forcément 4 est un multiple de 2...)
- On regarde si 5 est divisible par 5. Oui ! Donc on le divise par et on obtient le quotient 1 ; C'est fini !

Q9. Le jeu se joue avec 2 joueurs. Le but est de se rapprocher le plus près possible de 30 points sans les dépasser. Par exemple :

- Le joueur 1 lance le dé et obtient un 5. Donc on est à 5 points.
- Le joueur 2 lance le dé et obtient un 6. Donc on est à 11 points.
- Le joueur 1 lance le dé et obtient un 6. Donc on est à 17 points.
- Le joueur 2 lance le dé et obtient un 6. Donc on est à 23 points.

- Le joueur 1 lance le dé et obtient un 5. Donc on est à 28 points.
- Le joueur 2 décide de s'arrêter.
  Pour savoir qui gagne, on relance une dernière fois le dé. Si le nombre de points dépasse 30 points, alors c'est le joueur 2 qui gagne ; sinon c'est le joueur 1.