

OBJECTIFS : L'objectif principal de ce troisième TP est d'utiliser la structure « *if.. else* » dans des situations différentes.

DOCUMENT A RENDRE : Ce travail est évalué. Vous en rédigerez un compte-rendu numérique en utilisant un logiciel de traitement de texte (*Word ou Libre Office*). Le fichier constitué sera appelé *tp3.doc* ou *tp3.odt* et devra être transféré en fin d'activité **par l'intermédiaire** du site *nsibranly.fr* : se loguer et transférer en utilisant le code **tp3** . Ce compte-rendu contiendra :

- les réponses aux différentes questions posées,
- les captures d'écran **des morceaux de codes écrits et celles des résultats des exécutions** données dans le shell. Pour faire ces captures, utiliser *l'Outil Capture d'écran* de Windows.

1. Code pour les mathématiques :

Rappel de maths : Les diviseurs de 6 sont : 1 , 2 , 3 et 6 car le reste de la division euclidienne de 6 par 1 ou de 6 par 2 ou de 6 par 3 ou de 6 par 6 **est nul**.

Rappel pour python : Le reste de la division euclidienne de 6 par 2 par exemple se calcule en utilisant l'opérateur `%` . On a par exemple : `6 % 2 = 0` . Par contre `6 % 4 = 2` et donc 4 n'est pas un diviseur de 6.

Comment faire pour trouver les diviseurs d'un nombre n quelconque ? : On calcule le reste de la division euclidienne de n par 1, puis par 2, puis par , jusqu'à celui de n par n . Si ce reste est nul, on ajoute ce diviseur au message de sortie.

⇒ Compléter le code ci-dessous, qui demande un entier n à l'utilisateur et affiche en retour tous ses diviseurs.

En exécutant ce code pour $n = 6$ on obtiendra par exemple :

```
>>> (executing file "ex1.py")
Entre un entier : 6
Les diviseurs de 6 sont : 1 2 3 6
```

```
1 n = int(input("Entre un entier : "))
2 message = "Les diviseurs de "+str(n)+" sont : "
3
4
5
6
7
8
9 print(message)
```

2. Ecart entre 2 nombres :

⇒ Ecrire un programme qui demande 2 nombres à l'utilisateur et affiche en retour l'écart entre ces nombres (différence sans signe). Attention, le premier nombre donné ne sera pas nécessairement le plus petit des deux.

Exemples d'exécution :

```
>>> (executing file "ex2.py")
Premier nombre : 5
Second nombre : 1
Ecart : 4.0
```

```
>>> (executing file "ex2.py")
Premier nombre : 1
Second nombre : 5
Ecart : 4.0
```

3. Années bissextiles :

Info tirée de Wikipédia :

Depuis l'[ajustement du calendrier grégorien](#), l'année n'est bissextile (elle aura 366 jours)¹ que dans l'un des deux cas suivants :

1. si l'année est divisible par 4 et non divisible par 100 ;
2. si l'année est divisible par 400.

Dans un autre cas, l'année n'est pas bissextile : elle a la durée habituelle de 365 jours.

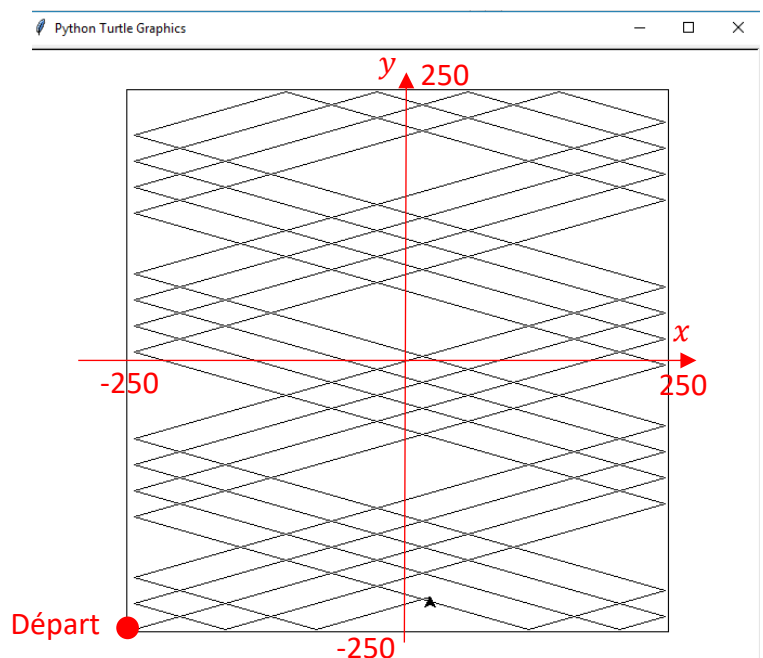
⇒ Ecrire un programme qui demande une année à l'utilisateur et indique s'il s'agit d'une année bissextile. Exemples d'exécution :

```
>>> (executing file "ex3.py")
Année : 2020
Année Bissextile
```

4. Tortue folle dans sa cage :

⇒ Ecrire un code qui :

- trace un carré centré sur l'origine du repère, de 500 px sur 500 px,
- positionne la tortue sur le point de départ de coordonnées (-250, -250)
- comporte une boucle « *for i in range(2000) :* » dans laquelle, la tortue se déplace à chaque itération d'une valeur $dx = 5$ et $dy = 2$. Si elle arrive au bord du carré, le signe de dx ou de dy est modifié ($dx = -dx$ ou $dy = -dy$) afin que la tortue puisse rebondir et repartir dans l'autre sens. Ainsi la tortue reste piégée dans sa cage.



Pour vous aider, écrire votre code en prenant modèle sur celui incomplet, donné ci-dessous :

```
1 from turtle import *
2
3 speed(10)
4 #Tracé d'un carré de coté a = 250 centré sur l'origine
5
6
7
8
9
10
11
12
13 #Position initiale de la tortue
14 up()
15 goto(-250,-250)
16 down()
17
18 #Déplacement tortue
19 dx = 5
20 dy = 2
21 for i in range(1000) :
22     #On récupère les coordonnées actuelles de la tortue
23     x,y = pos()
24     #On calcule les nouvelles coordonnées
25     x_new = x + dx
26     y_new = y + dy
27     #Si ces nouvelles coordonnées font sortir du carré,
28     #on change le signe du déplacement :
29     if x_new >= 250 or
30         dx = -dx
31         x_new = x + dx
32
33     if y_new >= 250 or y_new <= -250 :
34
35
36
37     #On déplace la tortue
38     goto(x_new , y_new)
39
40
41 exitonclick()
```

5. Modifier les lettres d'une chaîne de caractère :

a- Découverte d'une nouvelle structure utilisant « for » :

⇒ Exécuter le code ci-contre :

```
1 phrase = "Aujourd'hui c'est mardi"
2 for l in phrase :
3     print(l)
```

On voit que la structure « for » permet d'avoir accès à chacun des caractères de la chaîne de caractères contenue ici dans la variable « phrase ».

b- Utilisation de cette structure pour faire du cryptage :

⇒ Compléter le code ci-dessous, qui utilise cette structure, pour réécrire une phrase en remplaçant les « a » par des « e », les « i » par des « o » et les « u » par des « y ».

Exemples d'exécution:

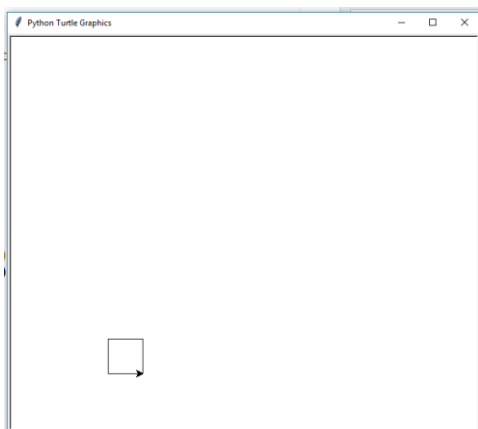
```
>>> (executing file "ex5.py")
eyjoyrd'hyo c'est merdo
```

```
1 phrase = "aujourd'hui c'est mardi"
2 phrase_modifiee = ""
3
4 for l in phrase :
5     if l == "a" :
6         phrase_modifiee = phrase_modifiee + "e"
7
8
9
10
11
12
13
14 print(phrase_modifiee)
```

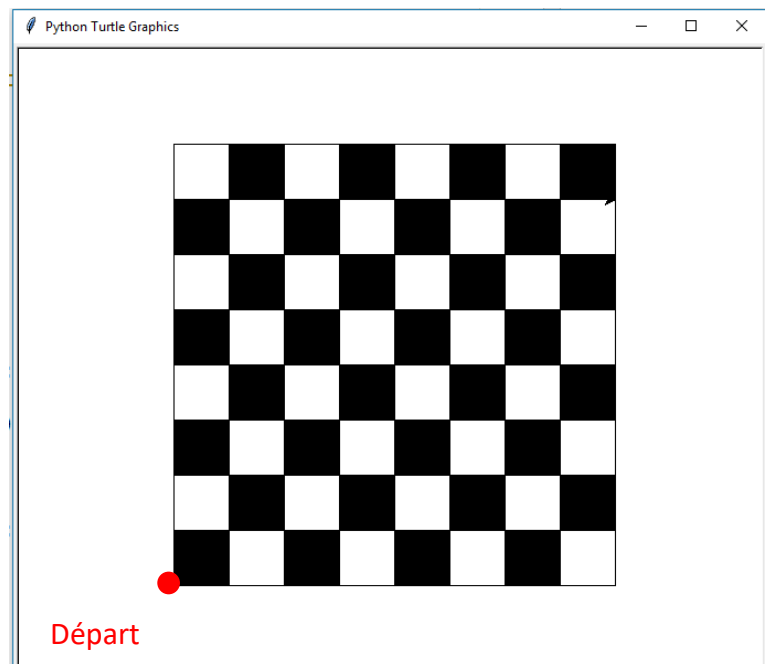
6. Créer un damier :

⇒ Créer un code qui crée le damier ci-contre, centré sur l'origine du repère et dont chaque case a une taille de 50px par 50px. Pour y arriver, procéder par étapes :

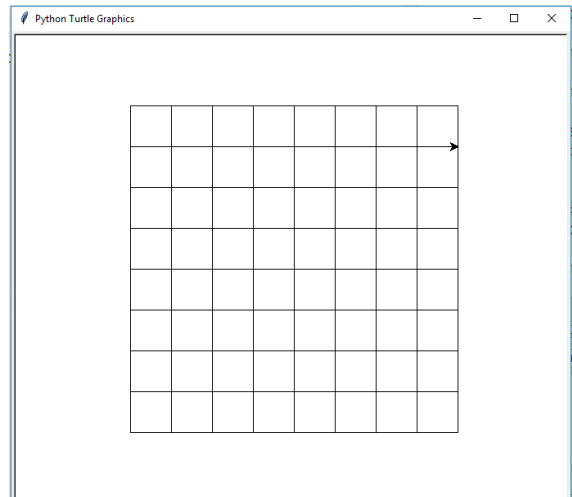
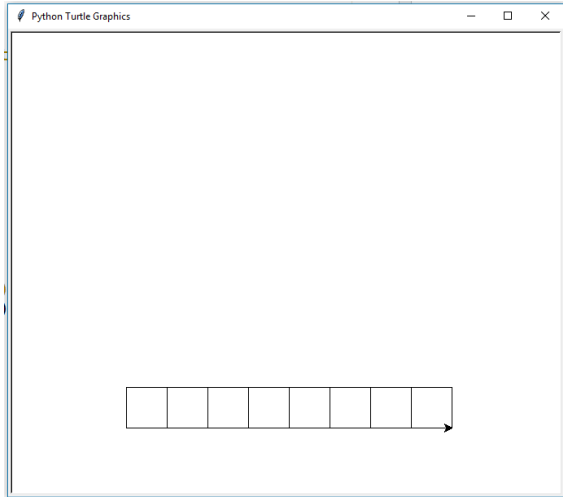
Etape 1:



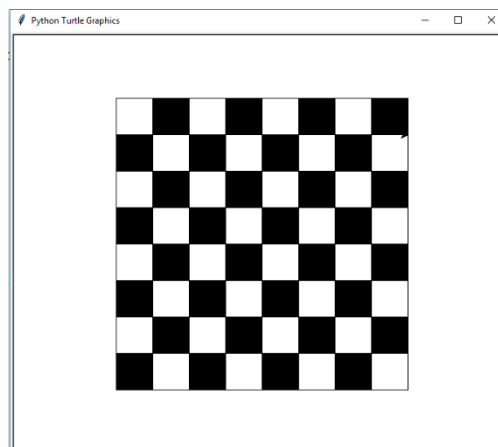
Etape 2:



Etape 3:



Etape 4 :



7. S'il vous reste du temps :

⇒ Reprendre l'exercice 4 de la tortue folle. Modifier le code pour que la tortue se déplace entre le carré de largeur 500px et un autre de largeur 400 px.

