

INFORMATIQUE II

LANGAGES - & - SYNTAXE PYTHON

EXERCICES

BLOCS CONDITIONNELS

Exercices introductifs

a - Guess a number !

Ecrire un programme qui demande à l'utilisateur de deviner un nombre entre 1 et 10, et qui affiche gagné si il choisit 7 !

b - Parité

Soit N un entier donné, écrire un bloc conditionnel qui affiche la parité de N.

c - Doomsday clock ! TicTacTicTac...

On donne les 2 dates suivantes Date1 = 2017-01-01 23:57:30 Date2 = 1953-01-01 23:58:00

- Trouver en Python laquelle est la plus ancienne ?

- Laquelle est la plus alarmiste puisque l'heure (dans la journée) indiquerait la fin du monde !

d - Jeu de couleur

On donne le programme suivant :

```
1 def arcEnCiel():
2     couleurArcEnCiel = ["rouge", "orange", "jaune", "vert", "bleu", "indigo", "violet"] #C1
3
4     mesCouleurs=[] #C2
5     maCouleur=""
6     while (maCouleur != "STOP"): #C3
7         maCouleur = input("Entrez une couleur en minuscules ou STOP pour arrêter :\n")#C4
8
9
10        if maCouleur in couleurArcEnCiel: #C5
11
12            if maCouleur not in mesCouleurs:#C6
13                mesCouleurs += [maCouleur] #C7
14            else:
15                print("Vous l'avez déjà dit !!!")#C8
16
17        elif maCouleur != "STOP":#C9
18            print(maCouleur + " ne fait pas partie des couleurs de l'arc en ciel !") #C10
19
20
21        print("Vous avez trouvé "+ str(len(mesCouleurs)) +" couleur(s) de l'arc en ciel :) #C11
22        print(mesCouleurs)
```

- A vous de le commenter entièrement !
- Décrire brièvement le déroulement du jeu.

Rq : vous pouvez le recopier mais le but est de comprendre le programme par simple lecture !

Exercices sur les blocs conditionnels

a - Positif ou négatif

Ecrire un programme qui demande un réel, et affiche s'il est «positif» ou «négatif».

Attention à prendre en compte le cas où le réel est nul !

b - Chaînes

Ecrire un programme qui demande une chaîne de caractères et affiche :

- les trois premiers caractères si le nombre de caractères est pair
- les trois derniers caractères si le nombre de caractères est impair
- qui renvoie un message d'erreur si ces opérations ne sont pas possibles.

c - Solution d'une équation polynomiale du second degré

Soient a , b et c les coefficients d'un polynôme du second degré

Faire un diagramme algorithmique qui détermine en fonction de a , b et c si il y a une ou deux solutions réelle(s) ou si il n'y en a pas.

Ecrire une fonction qui réalise cet algorithme :

- elle prend en argument a , b et c
- elle affiche le nombre de solution
- elle renvoie le tuple des solutions
( tuple vide si il n'y en a pas)

```
def solution2nd_DEG(a, b, c):  
    — Corps de fonction —  
    return solution
```

d - Détermination des années bissextiles (cf TP)

On admet, d'après le calendrier grégorien, qu'une année est bissextile si elle est divisible par 4, sauf si elle est divisible par 100 mais sauf si elle est aussi divisible par 400 !

- 1900, 2000, 2012 et 2014 sont elles bissextiles ?
- Faire un diagramme à Flèche pour représenter l'algorithme.
- Ecrire un programme qui prend en argument un entier N et renvoie la chaîne «bissextile» ou «non bissextile»
pour savoir si une année N est ou non bissextile.

e - Relation d'ordre dans le plan

Soient deux points dans le plan : P1 de coordonnées (x1, y1) et P2 de coordonnées (x2, y2).

On modélise ici un point par la donnée du tuple de ses coordonnées.

On admet la relation d'ordre suivante dans le plan :

$$P1 \geq P2 \quad \text{ssi} \quad \{ \quad x1 \geq x2 \quad \text{ou} \quad x1 = x2 \quad \text{et} \quad y1 \geq y2 \quad \}$$

- Faire un schéma pour illustrer cette définition : on prendra des points en exemple.
- On donne la fonction suivante qui prend en argument deux points et renvoie un booléen correspondant à la relation d'ordre ci-dessus :

```
1 def greaterThan(P1, P2):  
2  
3     #corps de fonction  
4  
5     return #resultat booléen
```

- A vous de compléter le corps de fonction.
- Vérifier votre fonction à l'aide des points exemples que vous aviez choisis.

f - Problème : Validité d'une date

On donne les dates par une chaîne de caractères de la forme suivante : «**2010-10-29 15:00:00**»
Cette forme en 19 caractères doit être respectée scrupuleusement pour que la date soit considérée comme valide.

On fera donc attention aux « - » , aux « : » mais aussi à la valeur des nombres et aux nombres de chiffres. Ex : pas de mois 00 ou de minute 60 et on note 03 (en 2 chiffres) le mois de mars !

Écrire une fonction **dateValide(date)** qui prend en argument une chaîne de caractère représentant la date et renvoie **True** sur la date est valide et **False** sinon.

Voilà des exemples de dates non valides pour vous guider : à vérifier !

2010-10-29- 15:00:00

2010-10-29 15-00-00

2010-10-29 15:00:00:

2010-10-29-15-00-00

2010-10-29-15:00:00

2010:10:29:15:00:00