

## Chapitre 2 : Alternatives - TD corrigé

### Exercice 1 : Trace de programme\*

Donner la trace du programme, c'est-à-dire donner l'affichage obtenu lors de l'exécution du programme.

```
In [1]: %load_ext tutormagic
```

```
In [2]: %%tutor -k
x = 12
print(x > 12)
print(x < 11 or (x > 40 and x < 100))
print(x != 9)
test = not(x > 10 and x <= 12) and x%2 == 0
print(test)
test = x>=10 and test
print(test)
test = not(test)
print(test)
test = not(test)
print(test)
```

[Python Tutor \(http://pythontutor.com/iframe-embed.html#code=x%20%3D%2012%20Aprint%28x%20%3E%2012%29%20Aprint%28x%20%3C%2011%20or%20%28x%20%3E%2040%20and%20x%20%3C%20100%29%29%20Aprint%28x%20%21%3D%209%29%20Atest%20%3D%20not%28x%20%3E%2010%20and%20x%20%3C%2012%29%20and%20x%252%20%3D%200%20Aprint%28test%29%20Atest%20%3D%20x%3E%3D10%20and%20test%20Aprint%28test%29%20Atest%20%3D%20not%28test%29%20Aprint%28test%29%20A&origin=opt-frontend.js&cumulative=false&heapPrimitives=false&textReferences=false&curlInstr=0&verticalStack=false&py=3&rawInputLstJSON=%5B%5D&codeDivWidth=50%25&codeDivHeight=100%25\)](http://pythontutor.com/iframe-embed.html#code=x%20%3D%2012%20Aprint%28x%20%3E%2012%29%20Aprint%28x%20%3C%2011%20or%20%28x%20%3E%2040%20and%20x%20%3C%20100%29%29%20Aprint%28x%20%21%3D%209%29%20Atest%20%3D%20not%28x%20%3E%2010%20and%20x%20%3C%2012%29%20and%20x%252%20%3D%200%20Aprint%28test%29%20Atest%20%3D%20x%3E%3D10%20and%20test%20Aprint%28test%29%20Atest%20%3D%20not%28test%29%20Aprint%28test%29%20A&origin=opt-frontend.js&cumulative=false&heapPrimitives=false&textReferences=false&curlInstr=0&verticalStack=false&py=3&rawInputLstJSON=%5B%5D&codeDivWidth=50%25&codeDivHeight=100%25))

### Exercice 2 : Affichage d'un test\*\*

1. Écrire un programme demandant à l'utilisateur de saisir un nombre entier. Si ce nombre est pair et positif ou s'il est impair et compris entre 5 (inclus) et 25 (inclus), alors le programme affiche `True`. Dans le cas contraire, le programme affiche `False`.

**Rappel :** Il existe un opérateur appelé *modulo* dont le symbole est `%` qui permet de calculer le reste de la division euclidienne. Par exemple, `3%2` vaut `1`.

1. Donner un jeu d'essai (4 tests significatifs) et prévoir les résultats.

```
In [ ]: #-----#
#   CORRECTION   ! ! ! ! !   #
#-----#
entier = int(input())
print( (entier%2==0 and entier > 0) or (entier%2==1 and entier>=5 and entier <=25))

# Jeux d'essai :
# 4 => affiche True car 4 est pair et supérieur à 0
# -4 => affiche False car -4 est pair mais pas positif
# 3 => affiche False car 3 est impair mais n'est pas compris entre 5 et 25
# 9 => affiche True car 9 est impair est compris entre 5 et 25
```

### Exercice 3 : Valeur absolue d'un nombre\*

Écrire un programme demandant à l'utilisateur de saisir un nombre et affichant la valeur absolue de ce nombre.

**Rappel :** La valeur absolue d'un nombre est sa valeur numérique sans son signe. Par exemple, la valeur absolue de -4 et de +4 est 4. Pour représenter la valeur absolue, on utilise la notation  $|\dots|$ . Ainsi, on écrit :  $|-4| = |+4| = 4$ .

```
In [2]: #-----#
#   CORRECTION   ! ! ! ! !   #
#-----#
print("Affichage de la valeur absolue")
print("Entrez un nombre :")
a = float(input())
res=a

if a<0:
    res = -a

print("|",a,"| = ",res)
```

```
Affichage de la valeur absolue
Entrez un nombre :
23
| 23.0 | = 23.0
```

### Exercice 4 : Le jeu des différences\*

Quelle différence y a-t-il entre les deux programmes suivants ?

```
In [ ]: A=5<2
if A:
    print("V")
else:
    print("F")
```

```
In [ ]: A=5<2
if A == True:
    print("V")
if A==False:
    print("F")
```

#### CORRECTION :

Les deux programmes produisent le même résultat mais il y a 3 tests élémentaires supplémentaires dans le second programme.

### Exercice 5 : Questions sur le `if` \*

#### Question 1 : Indentation

Qu'est-ce qu'une *indentation* ? Quel est l'intérêt d'une indentation ?

#### Question 2 : Else

Le `else` est-il obligatoire après un `if` ? Après un `if/elif` ?

#### Question 3 : Imbrication de `if`

Dans le programme suivant quelles instructions dépendent du premier `if` ? du deuxième `if` ?

```
In [ ]: print(1)
        if A>2:
            print(2)
            if B>=A:
                print(3)
            print(4)
        print(5)
```

Qu'affiche le programme dans les différents cas suivants ?

- quand  $A = 1$  et  $B = 6$ ,
- quand  $A = 4$  et  $B = 5$ ,
- quand  $A = 3$  et  $B = 0$ .

#### CORRECTION :

- Espaces insérés avant une instruction. L'indentation permet de définir un bloc d'instructions, associé par exemple à un `if`.
- Le `else` n'est obligatoire dans aucun des deux cas. On le met uniquement s'il y a des instructions à exécuter dans le cas où la condition (toutes les conditions si on a des `elif`) est fausse.
- Les lignes 3 à 6 forment le bloc d'instructions du premier `if` et la ligne 5 forme le bloc d'instructions du deuxième `if`.

Affichage obtenus suivants les différentes valeurs de  $A$  et  $B$  :

- 15
- 12345
- 1245

## Exercice 6 : Alternatives élémentaires\*

Pour quelle(s) valeur(s) de  $A$  l'instruction `print("B")` est-elle exécutée dans chacun des quatre codes ci-dessous?

```
In [ ]: if A > 10:
        print("A")
        else:
            print("B")
```

```
In [ ]: if A > 10:
        print("A")
        elif A > 200:
            print("B")
```

```
In [ ]: if A > 10:
        print("A")
        if A > 200:
            print("B")
```

```
In [ ]: if A > 10 and A < 10:
        print("A")
        else:
            print("B")
```

#### CORRECTION :

- si  $A \leq 10$
- jamais car  $A > 200$  n'est jamais vrai si  $A \leq 10$  (première condition fausse)
- Si  $A > 200$
- Pour n'importe quelle valeur (le test est toujours faux)

Pour quelle(s) valeur(s) de  $A$  l'instruction `print("C")` est-elle exécutée ?

```
In [ ]: if A < 100:
        print("A")
        elif A>=100:
        print("B")
        else:
        print("C")
```

#### CORRECTION :

Jamais car l'une des deux conditions est toujours vraie.

### Exercice 7 : Réduction du nombre de tests\*\*

Peut-on réduire le nombre de tests dans le programme suivant ? Si oui, comment ?

```
In [ ]: A=float(input())
        if A <= 10:
            print("A")
        elif A> 10 and A <= 50:
            print("B")
        elif A> 50 and A < 100:
            print("C")
        elif A >=100:
            print("D")
```

```
In [ ]: #-----#
        #  CORRECTION  ! ! ! ! !  #
        #-----#
        A=float(input())
        if A <= 10:
            print("A")
        elif A <= 50:
            print("B")
        elif A < 100:
            print("C")
        else :
            print("D")
```

## Pour aller plus loin

### Exercice 8 : Différence de deux nombres\*\*

Écrire un algorithme qui demande deux nombres entiers à l'utilisateur et calcule la différence entre le plus grand nombre et le plus petit nombre, quel que soit l'ordre de saisie.

```
In [ ]: #-----#
        #  CORRECTION  ! ! ! ! !  #
        #-----#
        print('Entrez le premier entier')
        entier1 = int(input())
        print('Entrez le deuxieme entier')
        entier2 = int(input())

        if entier1>entier2:
            resultat = entier1-entier2
        else:
            resultat = entier2-entier1

        print('La difference entre les deux entiers est ',resultat)
```

## Exercice 9 : Calcul de gabarit\*\*

On définit le `gabarit` d'un objet en fonction de sa `taille`. Le `gabarit` peut prendre les valeurs "Grand", "Moyen" ou "Petit" (qui sont des chaînes de caractères) selon que la `taille`, qui est un nombre entier, est respectivement supérieure ou égale à 10, comprise entre 4 (inclus) et 10 (non inclus) ou strictement inférieure à 4.

Écrire un programme qui demande à l'utilisateur la taille d'un objet et affiche après l'avoir déterminé le gabarit correspondant. On s'attachera à ne pas faire de test inutile.

```
In [ ]: #-----#
#  CORRECTION  ! ! ! ! !  #
#-----#
print("quelle est la taille de l'objet ?")
taille=int(input())

if taille < 4:
    gabarit = "Petit"
elif taille < 10 :
    gabarit = "Moyen"
else :
    gabarit = "Grand"
print("le gabarit est ",gabarit)
```

## Exercice 10 : Tri de trois nombres\*\*\*

Écrire un algorithme qui déclare 3 variables `a`, `b` et `c` initialisées avec 3 nombres entiers saisis par l'utilisateur. L'algorithme devra ensuite modifier les valeurs de `a`, `b` et `c` pour que `a` contienne la plus petite valeur saisie, `c` la plus grande valeur saisie et `b` la troisième valeur. L'algorithme affichera ensuite les valeurs de la plus petite à la plus grande.

```
In [ ]: #-----#
#  CORRECTION  ! ! ! ! !  #
#-----#
print('Tri de 3 nombres a,b et c par ordre croissant')

print('Donnez la valeur de a')
a = int(input())
print('Donnez la valeur de b')
b=int(input())
print('Donnez la valeur de c')
c= int(input())

if a>b:
    tmp = a
    a=b
    b=tmp

if b>c:
    tmp = c
    c=b
    b=tmp
    if a>b:
        tmp = a
        a=b
        b=tmp
print('a=',a, ' b=',b, ' c=',c)
```

## Exercice 11 : Racines d'un polynôme du second degré\*\*

Écrire un algorithme qui calcule et affiche les racines réelles d'un polynôme du second degré.

**NB :** Ce sont les racines réelles de l'équation  $ax^2 + bx + c = 0$

Prendre soin de bien analyser le problème et de savoir ce qui est très précisément demandé pour répertorier les données du problème.

**Remarque :** En Python, on obtient la racine carrée d'un nombre `x` grâce à l'appel `math.sqrt(x)`. Il faut auparavant avoir importé le module `math` grâce à l'instruction `import math`.

```
In [ ]: #-----#
#  CORRECTION  ! ! ! ! !  #
#-----#

import math
print("Calcul des racines reelles de a*x*x+b*x+c=0")

print("Donnez la valeur de a")
a = int(input())
print("Donnez la valeur de b")
b=int(input())
print("Donnez la valeur de c")
c= int(input())

#Distinction des cas
if a==0 and b==0 and c==0:    # 0x = 0
    print("Tout réel est une solution de cette équation")
elif a==0 and b==0:          # Contradiction: c # 0 et c = 0
    print("Cette équation ne possède pas de solutions.")
elif a==0:                    #bx + c = 0
    print("La solution de cette équation du premier degré est :")
    print("x = ",-c/b)
else :
    d = b*b-4*a*c
    if d<0:                    # b^2-4ac < 0
        print("Cette équation n'a pas de solutions réelles.")
    elif d==0:                 # b^2-4ac = 0
        print("Cette équation a une solution double réelle :")
        print(" x = ",-b/(2*a));
    else:                      # b^2-4ac > 0
        print("Deux solutions réelles :")
        print(" x1 =",(-b-math.sqrt(d))/(2*a))
        print(" x2 =",(-b+math.sqrt(d))/(2*a))
```