
Algorithmique et Programmation 1 – TD - TP 2

INSTRUCTIONS CONDITIONNELLES

CORRECTION

Exercice 1 - Equivalence d'expressions booléennes

Deux expressions booléennes sont équivalentes quand ces deux expressions s'évaluent toujours vers la même valeur booléenne pour toutes les valeurs possibles des variables qui apparaissent dans ces expressions.

Par exemple, pour x entier, $(x > 10 \text{ and } x < 12)$ est équivalente à $(x == 11)$. Par contre, $(x > y)$ et $(x! = y)$ ne sont pas équivalentes.

Dire si les expressions suivantes sont équivalentes pour x entier :

1. $(x > y \text{ or } x < y)$ et $(x! = y)$;
2. $(x! = 3 \text{ and } x! = 4 \text{ and } x! = 5)$ et $(x \leq 2 \text{ or } x \geq 6)$;
3. $(x == y \text{ and } x == z)$ et $(x == z)$;
4. $(x == y \text{ and } x == z)$ et $(x == y \text{ and } y == z)$

- | |
|---|
| <ol style="list-style-type: none">1. vrai2. vrai3. faux : si $x = 2$, $z = 2$ et $y = 3$, la première expression est fausse, tandis que la seconde est vraie4. vrai |
|---|

Exercice 2 - Simplification d'expressions booléennes

Simplifier une expression booléenne consiste à la remplacer par une autre expression booléenne équivalente plus simple.

Simplifier les expressions suivantes :

1. $x > 5 \text{ and } x > 7$
2. $x == y \text{ and } x == z \text{ and } y == z$
3. $x == 17 \text{ or } (x! = 17 \text{ and } x == 42)$
4. $x > 5 \text{ or } (x \leq 5 \text{ and } y > 5)$

- | |
|---|
| <ol style="list-style-type: none">1. $x > 7$2. $x == y \text{ and } x == z$3. $x == 17 \text{ or } x == 42$4. $x > 5 \text{ or } y > 5$ |
|---|

Exercice 3 - Différents tests

Donner les valeurs des variables à la fin des suites d'instructions suivantes.

1. Suite d'instruction 1 :

```
a = 2
b = a * a + 3
c = b - a
if (a == c) or (a < b):
    a = 2 * c
else:
    a = c - b
if (b < a) and (b > c):
    b = 2 * c + a
    c = 3 * a
if (c % b == 0) or (not(a < b) and not(a < c)):
    a, b = 0, 0
else:
    c = b - a
```

2. Suite d'instructions 2 :

```
a = 2
b = a * a + 3
c = b - a
if (a == c) or (a < b):
    a = 2 * c
else:
    a = c - b
    if (b < a) and (b > c):
        b = 2 * c + a
        c = 3 * a
if (c % b == 0) or (not(a < b) and not(a < c)):
    a, b = 0, 0
else:
    c = b - a
```

3. Suite d'instructions 3 :

```
a = 2
b = a * a + 3
c = b - a
if (a == c) or (a < b):
    a = 2 * c
elif (b < a) and (b > c):
    a = c - b
    b = 2 * c + a
    c = 3 * a
elif (c % b == 0) or (not(a < b) and not(a < c)):
    a, b = 0, 0
else:
    c = b - a
```

```

# -----
# Séquence 1 : 10 20 10
# -----
a = 2
b = a * a + 3      #b = 7
c = b - a          #c = 5
if (a == c) or (a < b): #True
    a = 2 * c      #a = 10
else :
    a = c - b
if (b < a) and (b > c): #True
    b = 2 * c + a    #b = 20
    c = 3 * a        #c = 30
if (c % b == 0) or (not (a < b) and not (a < c)): #False
    a, b = 0, 0
else:
    c = b - a

print("Sequence 1 :", a, b, c)

# -----
# Séquence 2 : 0 0 5
# -----
a = 2
b = a * a + 3      #b = 7
c = b - a          #c = 5
if (a == c) or (a < b): #True
    a = 2 * c      #a = 10
else :
    #on entre jamais dans le else
    a = c - b
    if (b < a) and (b > c):
        b = 2 * c + a
        c = 3 * a
if (c % b == 0) or (not (a < b) and not (a < c)): #True
    a, b = 0, 0
else:
    c = b - a

print("Sequence 2 :", a, b, c)

# -----
# Séquence 3 : 10 7 5
# -----
a = 2
b = a * a + 3      #b = 7
c = b - a          #c = 5
if (a == c) or (a < b): #True
    a = 2 * c      #a = 10
elif (b < a) and (b > c):      #on entre jamais dans les alternatives
    a = c - b
    b = 2 * c + a
    c = 3 * a
elif (c % b == 0) or (not (a < b) and not (a < c)):
    a, b = 0, 0
else:
    c = b - a

print("Sequence 3 :", a, b, c)

```

Exercice 4 - Simplification de code

Soit le programme suivant :

```
x = int(input("Donnez un entier x : "))
y = int(input("Donnez un entier y : "))

if ((x <= y) or (x >= y + 1)) :
    if ((x <= y) and (x < y)) :
        if (x < -x) :
            a = -x
            print(a)
        else :
            print(x)
    else :
        if (y * y * y < 0) :
            print(-y)
        else :
            print(y)
else :
    print(y)
```

1. Expliquer ce que fait ce programme
2. Simplifier ce programme

```
x = int(input("Donnez un entier x : "))
y = int(input("Donnez un entier y : "))

if ((x <= y) or (x >= y + 1)) : #toujours vrai pour des entiers
    if ((x <= y) and (x < y)) : #équivalent à x < y
        if (x < -x) : # équivalent à x < 0
            a = -x
            print(a) #équivalent à print(-x)
        else :
            print(x)
    else :
        if (y * y * y < 0) : #équivalent à y < 0
            print(-y)
        else :
            print(y)
else :
    print(y) #inutile

#Affiche la valeur absolue du plus petit entier

#Version simplifiée :

x = int(input("Donnez un entier x : "))
y = int(input("Donnez un entier y : "))

if (x < y) :
    if (x < 0) :
        print(-x)
    else :
        print(x)
else :
    if (y < 0) :
        print(-y)
    else :
        print(y)
```

Exercice 5 - Trois valeurs

Ecrire un programme Python qui demande à l'utilisateur de saisir trois valeurs entières, affecte la plus grande valeur à une variable `grand`, la valeur intermédiaire à une variable `moyen` et la plus petite des trois valeurs à une variable `petit`, puis affiche les valeurs de `grand`, `moyen` et `petit`.

```
nb1 = int(input("Entier 1 : "))
nb2 = int(input("Entier 2 : "))
nb3 = int(input("Entier 3 : "))

if (nb1 > nb2):
    if (nb1 > nb3):
        grand = nb1
        if (nb2 > nb3):
            moyen = nb2
            petit = nb3
        else:
            moyen = nb3
            petit = nb2
    else:
        grand = nb3
        moyen = nb1
        petit = nb2
else:
    if (nb2 > nb3):
        grand = nb2
        if (nb1 > nb3):
            moyen = nb1
            petit = nb3
        else:
            moyen = nb3
            petit = nb1
    else:
        grand = nb3
        moyen = nb2
        petit = nb1

print("Petit :", petit, "moyen :", moyen, "grand :", grand)
```

Exercice 6 - Division euclidienne

Ecrire un programme Python qui demande à l'utilisateur de saisir deux entiers positifs (le programme devra les stocker dans deux variables nommées dividende et diviseur); puis qui calcule le quotient et le reste de la division euclidienne de dividende par diviseur.

Attention, le programme devra vérifier que le dividende est positif ou nul, et que le diviseur est strictement plus grand que 0, et devra afficher un message d'erreur le cas échéant.

Voici quelques déroulés du programme attendu :

#Déroulé 1

Entrez le dividende (entier positif ou nul) : 9
Entrez le diviseur (entier strictement positif) : 2
Voici la division euclidienne de 9 par 2 :
 $9 = 2 * 4 + 1$

#Déroulé 2

Entrez le dividende (entier positif ou nul) : 9
Entrez le diviseur (entier strictement positif) : 0
Erreur, diviseur négatif ou nul

```
dividende = int(input("Entrez le dividende (entier positif ou nul) : "))

if dividende < 0:
    print("Erreur, dividende < 0")
else:
    diviseur = int(input("Entrez le diviseur (entier strictement positif) : "))
    if diviseur <= 0:
        print("Erreur, diviseur négatif ou nul")
    else:
        quotient = dividende//diviseur
        reste = dividende % diviseur
        print("Voici la division euclidienne de", dividende, "par", diviseur, ":")
        print(dividende, "=", diviseur, "*", quotient, "+", reste)
```

Exercice 7 - Convertir l'heure

Ecrire un programme Python qui demande à l'utilisateur un entier strictement positif représentant un nombre de secondes, et qui affiche une ligne exprimant la conversion de cet entier en heures-minutes-secondes. Votre programme doit afficher une erreur si le chiffre entré n'est pas >0

Voici quelques déroulés du programme attendu :

#Déroulé 1

Nombre de secondes (> 0) : 4872

4872 secondes correspond à 1 heure(s), 21 minute(s) et 12 seconde(s)

#Déroulé 2

Nombre de secondes (> 0) : -12

Erreur, -12 < 0

```
secondes = int(input("Nombre de secondes (> 0) : "))

if secondes < 0:
    print("Erreur,", secondes, "< 0")
else:
    heures = secondes // 3600
    reste_secondes = secondes % 3600

    minutes = reste_secondes // 60
    reste_secondes = reste_secondes % 60

    print(secondes, "secondes correspond à", heures, "heure(s)", minutes, "minute(s) et",
          reste_secondes, "seconde(s)")
```

Exercice 8 - Admissibilité d'un étudiant

Ecrire un programme Python qui vérifie l'admissibilité d'un étudiant qui passe trois examens. Il est déclaré admis si :

- soit, il a au moins 9 points à chaque examen.
- soit, la moyenne des trois examens est au moins égale à 10 points et la plus basse note est au moins égale 8 points.

S'il n'est pas admis alors il est refusé.

Voici un déroulé du programme attendu :

Veillez entrer le nom d'un étudiant ainsi ses trois notes

Entrer le nom de l'étudiant: Tom

Entrer la note 1: 18

Entrer la note 2: 6

Entrer la note 3: 15

Tom est refusé

```
# Admissibilité d'un étudiant

print("Veillez entrer le nom d'un étudiant ainsi ses trois notes")

nom_etudiant=input("Entrer le nom de l'étudiant: ")
note1=float(input("Entrer la note 1: "))
note2=float(input("Entrer la note 2: "))
note3=float(input("Entrer la note 3: "))

moyenne=(note1+note2+note3)/3
condition1= (note1>=9 and note2>=9 and note3>=9)
condition2= (moyenne>=10 and note1>8 and note2>=8 and note3>=8)

if condition1 or condition2:
    print(nom_etudiant,"est admis")
else:
    print(nom_etudiant,"est refusé")
```


Exercice 9 - Le plus grand en valeur absolue

Ecrire un programme Python qui demande à l'utilisateur deux chiffres, et affiche à l'écran le plus grand en valeur absolue. Si les deux nombres ont la même valeur absolue mais pas le même signe, le programme affichera celui qui est positif.

Voici quelques déroulés du programme attendu :

#Déroulé 1

Donnez un premier entier : -2
Donnez un second entier : 5
Le plus grand en valeur absolue est 5

#Déroulé 2

Donnez un premier entier : -5
Donnez un second entier : 2
Le plus grand en valeur absolue est -5

#Déroulé 3

Donnez un premier entier : -3
Donnez un second entier : 3
Le plus grand en valeur absolue est 3

#Déroulé 4

Donnez un premier entier : -3
Donnez un second entier : -3
Le plus grand en valeur absolue est -3

```
nb1 = int(input("Donnez un premier entier : "))
nb2 = int(input("Donnez un second entier : "))

if nb1 >= 0:
    nb1_abs = nb1
else:
    nb1_abs = -nb1

if nb2 >= 0:
    nb2_abs = nb2
else:
    nb2_abs = -nb2

if nb1_abs > nb2_abs:
    max = nb1
elif nb1_abs < nb2_abs:
    max = nb2
elif nb1 != nb2:
    max = nb1_abs
else:
    max = nb1

print("Le plus grand en valeur absolue est", max)
```

Exercice 10 - Cinéma

Un cinéma propose les tarifs suivants pour les groupes :

- 8€ la place pour les 5 premières ;
- 6€ la place pour les suivantes jusqu'à 10
- 5,50€ la place, celles au-delà de 10

Ecrire un programme Python qui affiche le prix total pour un nombre de tickets entré au clavier.

```
places = int(input("Combien de places souhaitez vous ? "))

prix = 0

if places >= 10:
    prix = (places - 10) * 5.5 + 6 * 5 + 8 * 5
elif places > 5 and places <= 10:
    prix = (places - 5) * 6 + 8 * 5
elif places > 0 and places <= 5:
    prix = places * 8

print("Pour", places, "places, vous devez", prix, "euros")
```

Exercice 11 - Ordinaux anglais

On s'intéresse aux ordinaux anglais abrégés, où le nombre est écrit en chiffre : 1st, 2nd, 3rd, 4th, ..., 10th, 11th, 12th, ..., 20th, 21st, 22nd, 23rd, ...

Pour déterminer le suffixe, on regarde le dernier chiffre du nombre : si c'est 1, on ajoute le suffixe st ; si c'est 2, le suffixe est nd ; si c'est 3, le suffixe est rd ; sinon, le suffixe est th. Il y a cependant une exception : si l'avant dernier chiffre est 1, le suffixe est toujours th.

Ecrire un programme qui demande un entier à l'utilisateur, et affiche l'ordinal anglais abrégé correspondant.

```
n = int(input("Donnez un entier : "))

if (4 <= (n % 100)) and ((n % 100) <= 20) :
    print(n, "th", sep = "")
elif (n - 1) % 10 == 0 :
    print(n, "st", sep = "")
elif (n - 2) % 10 == 0 :
    print(n, "nd", sep = "")
elif (n - 3) % 10 == 0 :
    print(n, "rd", sep = "")
else :
    print(n, "th", sep = "")
```