



DU Développeur II

Algorithmique et Python

Jean-Luc.Bourdon@u-cergy.fr

TP n°02 (correction)
Structures conditionnelles

RAPPEL : Pour **chacun des exercices**, vous écrirez la réponse **sous forme de pseudo-code** (algorithme) et vous la coderez **en langage Python** (programme). Pour ce dernier, vous créez donc **un fichier Python** que vous exécuterez depuis votre terminal Linux.

Exercice 1 : minimum

- Écrivez un algorithme qui va prendre en entrée 2 valeurs entières
- En sortie, cet algorithme va afficher la plus petite valeur entre les 2

Algorithme attendu

```
1 #
2 # L1 MIPI S1 2021-2022 : TP 2, exercice 1, algo.
3 #
4
5 programme principal
6 variables
7     entrée
8         entier val1, val2
9     sortie
10        entier mini
11 début
12     écrire("Entrez_la_valeur_entière_1:_")
13     lire(val1)
14     écrire("Entrez_la_valeur_entière_2:_")
15     lire(val2)
16     mini <- val1
17     si (val2 < val1) alors
18         mini <- val2
19     fin_si
20     écrire("La_valeur_minimale_entre_val1={val1}_et_val2={val2}_est_{mini}")
21 fin
```

Code Python attendu

```
1 #
2 # L1 MIPI S1 2021-2022 : TP n°2, exercice 1, Python
3 #
4 ## Récupérations des valeurs d'entrée
5 print("Entrez_la_valeur_entière_1:_")
6 val1 = int(input())
7 print("Entrez_la_valeur_entière_2:_")
8 val2 = int(input())
9 ## initialisation de mini avec val1
10 mini = val1
11 ## test si val2 est plus petite que val1
12 if (val2 < val1) : ### val2 < val1
13     mini = val2
14 ## affichage du résultat
15 print(f"La_valeur_minimale_entre_val1={val1}_et_val2={val2}_est_{mini}")
```

Exercice 2 : pair / impair

- Écrivez un algorithme qui va demander à l'utilisateur de saisir une valeur entière
- En sortie, on stockera dans une variable booléenne si la valeur d'entrée est paire ou impaire
- Utilisez l'opérateur 'modulo' pour réaliser cet exercice
- Affichez un message correspondant au résultat
- Une fois l'algorithme et le programme faits, modifiez les en utilisant la division entière au lieu du 'modulo'

Algorithme attendu

```
1 #
2 # L1 MIPI S1 2021-2022 : TP 2, exercice 2, algo.
3 #
4
5 programme principal
6 variables
7     entrée
8     entier val
9     sortie
10    booleen isEven
11 début
12     écrire("Entrez une valeur entière : ")
13     lire(val)
14     isEven <- val % 2
15     si (isEven = VRAI) alors
16         écrire("La valeur {val} est paire")
17     sinon
18         écrire("La valeur {val} est impaire")
19     fin_si
20 fin
```

Code Python attendu

```
1 #
2 # L1 MIPI S1 2021-2022 : TP n°2, exercice 2, Python
3 #
4
5 ### récupération de la valeur d'entrée
6 print("Entrez une valeur entière : ")
7 val = int( input() )
8
9 ### calcul du booléen
10 isEven = val % 2
11
12 ### test du booléen et affichage
13 if (isEven == True): ### isEven == True
14     print("La valeur {val} est paire")
15 else:
16     print("La valeur {val} est impaire")
```

Exercice 3 : résultat d'examen

- Écrivez un algorithme qui va demander à l'utilisateur de saisir une valeur réelle entre 0 et 20
- Testez la valeur et affichez un message en fonction :
 - "Ajourné" si la valeur est inférieure ou égale à 6
 - "Rattrapage" si la valeur est supérieure à 6 et inférieure à 10
 - "Admis" si la valeur est supérieure ou égale à 10
 - Un message d'erreur si la note est en dehors de l'intervalle [0,20]

Algorithme attendu

```
1  #
2  # L1 MIPI S1 2021-2022 : TP 2, exercice 3, algo.
3  #
4
5  programme principal
6  variables
7      entrée
8      entier note
9  début
10     écrire("Entrez_une_note_entière:_")
11     lire(note)
12     si (note < 0) OU (note > 20) alors
13         écrire("Entrez_une_note_entre_0_et_20")
14     sinon si (note <= 6) alors
15         écrire("Ajourné_avec_note={note}_")
16     sinon si (note < 10) alors
17         écrire("Rattrapage_avec_note={note}_")
18     sinon
19         écrire("Admis_avec_note={note}_")
20     fin_si
21 fin
```

Code Python attendu

```
1  #
2  # L1 MIPI S1 2021-2022 : TP n°2, exercice 3, Python
3  #
4
5  ## donnée d'entrée
6  print("Entrez_une_note_entière:_")
7  note = int( input() )
8
9  ## test de la valeur de la note par tranche, et affichages correspondants
10 if ((note < 0) or (note > 20)):    ### note < 0 or note > 2
11     print("Entrez_une_note_entre_0_et_20")
12 elif (note <= 6):                ### note <= 6
13     print(f"Ajourné_avec_note={note}_")
14 elif (note < 10):                ### note < 10
15     print(f"Rattrapage_avec_note={note}_")
16 else:
17     print(f"Admis_avec_note={note}_")
```

Exercice 4 : équation de droite

Soit une droite d'équation : $4x + 2y + 6 = 0$

- Écrivez un algorithme qui va demander à l'utilisateur de saisir les valeurs réelles x et y
- Déterminez par calcul si le point de coordonnées $[x,y]$ est situé en-dessous de la droite, au-dessus ou bien exactement dessus
- Affichez un message correspondant au résultat

Algorithme attendu

```
1 #
2 # L1 MIPI S1 2021-2022 : TP 2, exercice 4, algo.
3 #
4
5 programme principal
6 variables
7     entrée
8         réel x, y
9     sortie
10        booléen res
11 début
12     écrire("Entrez_la_valeur_réelle_x:_")
13     lire(x)
14     écrire("Entrez_la_valeur_réelle_y:_")
15     lire(y)
16     res  $\leftarrow 4*x + 2*y + 6$ 
17     si (res < 0) alors
18         écrire("Le_point_x={x}/y={y}_est_en-dessous_de_la_droite")
19     sinon si (res > 0) alors
20         écrire("Le_point_x={x}/y={y}_est_au-dessus_de_la_droite")
21     sinon
22         écrire("Le_point_x={x}/y={y}_est_situé_exactement_sur_la_droite")
23     fin_si
24 fin
```

Code Python attendu

```
1 #
2 # L1 MIPI S1 2021-2022 : TP n°2, exercice 4, Python
3 #
4
5 ## données d'entrée
6 print("Entrez_la_valeur_réelle_de_x:_")
7 x = float(input())
8 print("Entrez_la_valeur_réelle_de_y:_")
9 y = float(input())
10
11 ## Calcul
12 res = 4*x + 2*y + 6
13
14 ## test de la valeur du calcul et affichage
15 if (res < 0):    ### res < 0
16     print(f"Le_point_x={x}/y={y}_est_en-dessous_de_la_droite")
17 elif (res > 0):    ### res > 0
18     print(f"Le_point_x={x}/y={y}_est_au-dessus_de_la_droite")
19 else:
20     print(f"Le_point_x={x}/y={y}_est_situé_exactement_sur_la_droite")
```

Exercice 5 : signe d'un produit

- Écrivez un algorithme qui va demander à l'utilisateur de saisir 2 valeurs entières
- Déterminez par calcul si le produit de ces 2 variables est positif, négatif ou nul
- Attention : pour cet exercice il est interdit d'utiliser l'opérateur de multiplication. Vous devez déterminer le résultat d'une autre manière
- Affichez un message correspondant au résultat

Algorithme attendu

```
1 #
2 # L1 MIPI S1 2021-2022 : TP 2, exercice 5, algo.
3 #
4
5 programme principal
6 variables
7     entrée
8         entier a, b
9     interne
10         booléen isPositiveA
11         booléen isPositiveB
12         booléen isZero
13 début
14     écrire("Entrez_la_valeur_entière_1:_")
15     lire(a)
16     écrire("Entrez_la_valeur_entière_2:_")
17     lire(b)
18     isZero <- (a = 0) ou (b = 0)
19     isPositiveA <- (a > 0)
20     isPositiveB <- (b > 0)
21     si (isZero = VRAI) alors
22         écrire("Le_produit_de_{a}x{b}_est_nul")
23     sinon si (isPositiveA = isPositiveB) alors
24         écrire("Le_produit_de_{a}x{b}_est_positif")
25     sinon
26         écrire("Le_produit_de_{a}x{b}_est_négatif")
27     fin_si
28 fin
```

Code Python attendu

```
1 #
2 # L1 MIPI S1 2021-2022 : TP n°2, exercice 5, Python
3 #
4
5 ## récupération des valeurs d'entrée
6 print("Entrez_la_valeur_entière_1:_")
7 a = int( input() )
8 print("Entrez_la_valeur_entière_2:_")
9 b = int( input() )
10
11 ## calcul des booléens internes
12 isZero = (a == 0) or (b == 0)
13 isPositiveA = (a > 0)
14 isPositiveB = (b > 0)
15
16 ## test des booléens et affichage
17 if (isZero == True): ### isZero == True
18     print(f"Le_produit_de_{a}x{b}_est_nul")
19 elif (isPositiveA == isPositiveB): ### isPositiveA == isPositiveB
20     print(f"Le_produit_de_{a}x{b}_est_positif")
21 else:
22     print(f"Le_produit_de_{a}x{b}_est_négatif")
```

Exercice 6 : offre promotionnelle

- Soit un magasin qui vend des masques à l'unité avec une offre promotionnelle
- Les 20 premiers masques achetés coûtent 0,20 Euros chaque
- Les 10 masques suivants coûtent 0,15 Euros chaque
- Tout masque supplémentaire sera facturé 0,10 Euros chaque
- Écrivez un algorithme qui va demander à l'utilisateur de saisir le nombre de masques voulu, et qui déterminera le montant total à payer pour cette quantité
- Affichez le résultat

Algorithme attendu

```
1  #
2  # L1 MIPI S1 2021-2022 : TP 2, exercice 6, algo.
3  #
4
5  programme principal
6  variables
7      entrée
8          entier n
9      sortie
10         réel prix ← 0
11  début
12      écrire ("Entrez le nombre de masques : ")
13      lire(n)
14      si (n < 0) alors
15          n ← 0
16      sinon si (n ≤ 20) alors
17          prix ← prix + (n * 0.2)
18      sinon si (n ≤ 30) alors
19          prix ← prix + (20 * 0.2)
20          prix ← prix + ((n-20) * 0.15)
21      sinon
22          prix ← prix + (20 * 0.2)
23          prix ← prix + (10 * 0.15)
24          prix ← prix + ((n-30) * 0.1)
25  fin_si
26      écrire ("Le prix total pour {n} masques est de {prix} euros.")
27  fin
```

Code Python attendu

```
1 #
2 # L1 MIPI S1 2021-2022 : TP n°2, exercice 6, Python
3 #
4
5 ## initialisation du prix de sortie
6 prix = 0
7
8 ## saisie du nombre de masques en entrée
9 print("Entrez_le_nombre_de_masques:_")
10 n = int( input() )
11
12 ## test du nombre de masques
13 if (n < 0):
14     ## si n est négatif, on force la valeur n à 0
15     n=0
16 elif (n <= 20):
17     ## le nombre de masques est entre 0 et 20 inclus
18     prix = prix + (n * 0.2)
19 elif (n <= 30):
20     ## le nombre de masques est entre 21 et 30 inclus
21     prix = prix + (20 * 0.2)
22     prix = prix + ((n-20) * 0.15)
23 else:
24     ## le nombre de masques est au-delà de 30
25     prix = prix + (20 * 0.2 )
26     prix = prix + (10 * 0.15)
27     prix = prix + ((n-30) * 0.1)
28
29 ## On affiche le résultat du calcul
30 print(f"Le_prix_total_pour_{n}_masques_est_de_{prix}_euros.")
```

Exercice 7 : Affichage d'une durée

- Écrivez un algorithme qui va demander à l'utilisateur de saisir 3 valeurs entières correspondant à une durée (nombre d'heures, nombre de minutes, nombre de secondes)
- L'algorithme doit mettre à jour ces 3 valeurs si on ajoute 1 seconde à la durée totale
- Il faudra donc penser à gérer le changement de minute, ainsi que le changement d'heure
- Affichez la durée mise à jour en fin d'algorithme (affichez les heures, minutes, secondes)

Algorithme attendu

```
1  #
2  # L1 MIPI S1 2021-2022 : TP 2, exercice 7, algo.
3  #
4
5  programme principal
6  variables
7      entrée/sortie
8          entier H, M, S
9      interne
10         booléen hourOK <- VRAI
11         booléen minOK  <- VRAI
12         booléen secOK  <- VRAI
13  début
14     écrire("Entrez_les_heures_: ")
15     lire(H)
16     écrire("Entrez_les_minutes_[0-59]: ")
17     lire(M)
18     écrire("Entrez_les_secondes_[0-59]: ")
19     lire(S)
20     hourOK <- (H >= 0)
21     minOK  <- (M >= 0) ET (M < 60)
22     secOK  <- (S >= 0) ET (S < 60)
23     si (hourOK ET minOK ET secOK) alors
24         écrire("L'heure_est_{H}:{M}:{S}")
25         S <- S + 1
26         si (S = 60) alors
27             S <- 0
28             M <- M+1
29         fin_si
30         si (M = 60) alors
31             M <- 0
32             H <- H+1
33         fin_si
34         écrire("La_nouvelle_heure_après_1_seconde_est_{H}:{M}:{S}")
35     sinon
36         écrire("Vous_avez_entré_des_données_erronées")
37     fin_si
38  fin
```


Code Python attendu

```
1 #
2 # L1 MIPI S1 2021-2022 : TP n°2, exercice 7, Python
3 #
4
5 ## initialisation des variables internes
6 hourOK = True
7 minOK  = True
8 secOK  = True
9
10 ## saisie des valeurs d'entrée
11 print("Entrez les heures:_")
12 H = int( input() )
13 print("Entrez les minutes_[0-59]:_")
14 M = int( input() )
15 print("Entrez les secondes_[0-59]_:_" )
16 S = int( input() )
17
18 ## mise a jour des variables internes
19 hourOK = (H >= 0)
20 minOK  = (M >= 0) and (M < 60)
21 secOK  = (S >= 0) and (S < 60)
22
23 ## test des variables booléennes internes
24 if (hourOK and minOK and secOK):
25     ## affichage de l'heure initiale
26     print(f"L'heure_est_{H}:{M}:{S}")
27     ## incrément du nombre de secondes
28     S = S+1
29     ## test du nombre de secondes qui dépasse 59
30     if (S == 60):
31         S = 0
32         M = M+1
33     ## test du nombre de minutes qui dépasse 59
34     if M == 60:
35         M = 0
36         H = H+1
37     print(f"La_nouvelle_heure_après_1_seconde_est_{H}:{M}:{S}")
38 else:
39     print("Vous_avez_entré_des_données_erronnées")
```