



1 Concevoir une séquence d'instructions conditionnelles

Énoncé Une compagnie de transports en commun propose à ses usagers deux modes de tarification.

Tarif A : 1,50 € le trajet.

Tarif B : 20 € par mois, puis 0,80 € le trajet.

1. L'algorithme ci-contre permet d'indiquer le tarif le plus avantageux suivant le nombre de trajets n effectués dans le mois, n étant un entier entré par l'utilisateur.

Compléter cet algorithme en précisant la signification des variables A et B .

2. Programmer cet algorithme dans le langage de votre choix, puis le tester pour différentes valeurs de n .

3. À partir de combien de trajets le tarif B est-il plus intéressant ? Justifier.

```
A ← 1,5 × n
B ← ...
Si ... Alors
    Afficher « Le tarif A est le plus intéressant. »
Sinon
    Afficher « Le tarif B est le plus intéressant. »
Fin Si
```

Solution

1. n représente le nombre de trajets effectués dans le mois. La variable A (resp. B) représente le prix d'achat pour n trajets mensuels au tarif A (resp. au tarif B). On complète l'affectation $B \leftarrow 20 + 0,80 \times n$.

2. Voici les programmes dans les langages Casio, TI et Python.

```
Casio
"NB TRAjets"?)>N#
1.5N#A#
20#0.8N#B#
If A#B#
Then #
"Tarif A +interess"#
Else #
"tarif B +interess"#
IEnd#

TI
Input "NB TRAJE" N
1.5N#A#
20#0.8N#B#
If A#B#
Then #
Disp "TARIF A +
INTERESSANT"
Else #
Disp "TARIF B +
INTERESSANT"
End
```

```
Python
1n=int(input("Nombre de trajets ?"))
2A=1.5*n
3B=20+0.8*n
4if A<B:
5    print("Le tarif A est le plus intéressant.")
6else:
7    print("Le tarif B est le plus intéressant.")
```

Point méthode
1. Il est parfois nécessaire de connaître le type d'une variable. Ici n est de type *nombre entier*, A et B sont de type *nombre réel*.

Point méthode
2. Les erreurs de syntaxes sont courantes et sont indiquées lors de l'exécution du programme. Attention, l'absence d'erreurs de syntaxe ne garantit pas que le programme est correct. Il est utile de le tester pour plusieurs valeurs de n .

3. Pour n trajets effectués, le prix de revient avec le tarif A (resp. B) est 1,5 n

3. Pour n trajets effectués, le prix de revient avec le tarif A (resp. B) est 1,5 n (resp. 0,8 n + 20). Le tarif B est plus intéressant que le A si et seulement si :

$$1,5n \geq 0,8n + 20 \Leftrightarrow 0,7n \geq 20 \Leftrightarrow n \geq \frac{20}{0,7} \Leftrightarrow n \geq 29 \text{ (car } n \text{ est un entier).}$$

À partir de 29 tickets, le tarif B est plus intéressant que le tarif A .



2 Lire, comprendre, modifier ou compléter un algorithme ou un programme

Énoncé Dans le plan muni d'un repère orthonormé, on considère les points $A(x_A; y_A)$ et $B(x_B; y_B)$.

1. a. Exprimer le carré AB^2 de la distance AB en fonction des coordonnées x_A, y_A, x_B et y_B .
b. Programmer en langage Python une fonction `carre_dist` qui prend comme paramètres les coordonnées des points A et B et qui renvoie le carré de la distance AB .

2. Trois points A, B et C étant définis par leurs coordonnées, on se demande si le triangle ABC est isocèle. On souhaite résoudre ce problème en faisant appel à la fonction `carre_dist` et en utilisant la fonction `est_isocèle` incomplètement reproduite ci-dessous. La compléter.

```
Fonction est_isocèle(xA,yA,xB,yB,xC,yC)
    Si carre_dist(xA,yA,xB,yB) = ... Alors Afficher(« le triangle est isocèle en A ») Fin Si
    Si carre_dist(xA,yA,xB,yB) = ... Alors Afficher(« le triangle est isocèle en B ») Fin Si
    Si ... = ... Alors Afficher(« le triangle est isocèle en C ») Fin Si
Fin Fonction
```

3. Programmer en langage Python la fonction `est_isocèle`, puis déterminer si le triangle ABC est isocèle ou non, dans chacun des cas suivants.

- a. $A(-6; 9); B(3; 11); C(1; 2)$
- b. $A(4; -1); B(2; 5); C(-1; 3)$

Solution

1. a. $AB^2 = (x_B - x_A)^2 + (y_B - y_A)^2$

↳ Voir Chap. 6 p. 164

b. `def carre_dist(xA,yA,xB,yB):`
 `return (xB-xA)**2+(yB-yA)**2`

2. On obtient :

```
Fonction est_isocèle(xA,yA,xB,yB,xC,yC)
    Si carre_dist(xA,yA,xB,yB) = carre_dist(xA,yA,xC,yC) Alors
        Afficher(« le triangle est isocèle en A ») Fin Si
    Si carre_dist(xA,yA,xB,yB) = carre_dist(xB,yB,xC,yC) Alors
        Afficher(« le triangle est isocèle en B ») Fin Si
    Si carre_dist(xA,yA,xC,yC) = carre_dist(xB,yB,xC,yC) Alors
        Afficher(« le triangle est isocèle en C ») Fin Si
Fin Fonction
```

Point méthode
2. Pour nommer une fonction, il est intéressant d'utiliser un nom qui explicite ses caractéristiques. Ici, on a choisi l'adjectif *isocèle* et les coordonnées des sommets du triangle examiné, qui sont les paramètres utilisés par la fonction.

3. Cette fonction se traduit en langage Python par :

```
def est_isocèle(xA,yA,xB,yB,xC,yC):
    if carre_dist(xA,yA,xB,yB)==carre_dist(xA,yA,xC,yC):
        print("Le triangle ABC est isocèle en A.")
    if carre_dist(xA,yA,xB,yB)==carre_dist(xB,yB,xC,yC):
        print("Le triangle ABC est isocèle en B.")
    if carre_dist(xA,yA,xC,yC)==carre_dist(xB,yB,xC,yC):
        print("Le triangle ABC est isocèle en C.")
```

- a. Le triangle ABC est isocèle en B .
- b. L'exécution du programme n'affiche rien : ce triangle n'est pas isocèle.

Énoncé Tom dispose de 30 jetons. Il en pose un sur la 1^{re} case d'un échiquier, puis deux sur la 2^e, trois sur la 3^e et ainsi de suite.

1. a. On souhaite appliquer l'algorithme ① ci-contre avec $n = 3$.

Compléter un tableau d'état des variables au fur et à mesure de l'exécution de l'algorithme.

Quelle est la valeur finale de la variable S ?

b. Comment utiliser cet algorithme dans le contexte de l'exercice ?

2. En construisant un tableau similaire au précédent, appliquer l'algorithme ②.

Quelle est la valeur finale de la variable k ?

Que représente-t-elle dans le contexte de l'exercice ?

```
S ← 0
Pour k variant de 1 à n Faire
  S ← S + k
Fin Pour
```

```
S ← 0
k ← 0
Tant que S ≤ 30 Faire
  k ← k + 1
  S ← S + k
Fin Tant que
```

Solution

1. a. D'après le tableau d'état des variables, la valeur finale de S est 6.

Étapes	Valeurs de n	Valeurs de k	Valeurs de S
Initialisation	3	-	0
1 ^{er} passage	3	1	0+1=1
2 ^e passage	3	2	1+2=3
Dernier passage	3	3	3+3=6

b. Cet algorithme calcule la somme des entiers consécutifs de 1 à n . Ainsi, S contient au final la valeur $1+2+3+\dots+n$. Il s'agit du nombre total de jetons posés sur les n premières cases de l'échiquier.

2. D'après le tableau d'état des variables, la valeur finale de k est 8.

Étapes	Valeurs de k	Valeurs de S	Test $S \leq 30$
Initialisation	0	0	Vrai
1 ^{er} passage	1	0+1=1	Vrai
2 ^e passage	2	1+2=3	Vrai
⋮	⋮	⋮	⋮
7 ^e passage	7	21+7=28	Vrai
8 ^e passage	8	28+8=36	Faux : sortie de boucle

Lorsque l'on dispose de trente jetons, on ne peut empiler des jetons de cette manière que sur les 8 premières cases de l'échiquier.

Point méthode

1. a. On applique pas à pas les instructions de l'algorithme en indiquant, à chaque étape, la valeur courante de chacune des variables de l'algorithme. Ce tableau permet de bien comprendre le fonctionnement de l'algorithme et comment il agit sur chacune des variables mises en jeu.

Point méthode

2. Dans le cas des boucles *Tant que*, on peut indiquer, dans une colonne, le résultat (Vrai ou Faux) du test apparaissant dans la condition (ici $S \leq 30$). Cela permettra de connaître le moment où on sort de la boucle *Tant que*.

J'applique

1 Un imprimeur élabore des cartes de visite et propose à ses clients une facturation dégressive.

Le tarif est de 0,2 € par carte de visite jusqu'à la centième unité, puis de 0,15 € par carte supplémentaire.

1. Calculer le prix facturé pour 50 cartes de visite commandées, puis celui pour 150 cartes commandées.

2. Le programme Python ci-dessous affiche le tarif facturé pour x cartes de visite commandées.

Compléter ce programme, puis le tester.

```
1 x=int(input("Nombre de cartes commandées ?"))
2
3 if ... :
4   tarif=0.2*x
5 else:
6   tarif= ...
7
8 print(tarif)
```

2 Élara a une liste de 10 triangles dont elle doit déterminer s'ils sont rectangles à partir des longueurs de leurs trois côtés. Elle décide d'utiliser le langage Python pour arriver à ses fins.

1. Le triangle dont les côtés mesurent $a = 5$, $b = 9$ et $c = 8$ est-il rectangle ?

2. a. Combien de variables sont nécessaires à l'écriture de ce programme ? De quel type sont-elles ?

b. Lorsque le triangle est rectangle, Élara souhaite que a soit l'hypoténuse. Compléter le programme Python ci-dessous pour que a contienne la plus grande valeur.

NumWorks

```
from math import *
a,b,c=4,5,3
if b<c:
  a,b,c=b,a,c
else:
  a,b,c=c,b,a
elif a<b:
  a,b,c=b,a,a
else:
  a,b,c=b,a,c
```

J'applique

3 La fonction ci-contre est écrite en Python.

1. Que renvoie-t-elle lorsque x vaut 3 ?

2. On saisit $f(1)$ dans la console. Qu'obtient-on ?

```
def f(x):
  if x>2:
    return x**2
  else:
    return 2*x
```

4 **1.** Compléter la fonction ci-contre pour qu'elle renvoie le maximum des trois valeurs passées en argument.

2. Tester la fonction avec les arguments $a=3$, $b=3$ et $c=3$.

```
def max(a,b,c):
  if a>b:
    if a>c:
      return a
    else:
      ...
```

J'applique

5 Modifier, programmer puis exécuter l'algorithme ② de l'exercice ci-dessus afin de déterminer le plus petit entier n pour lequel la somme $S = 1 + 2 + 3 + \dots + n$ dépasse 1 000.

6 Une fourmilière compte 200 fourmis. Chaque année, cette population augmente de 5 %.

On souhaite étudier son évolution sur 10 ans. Pour cela, on élabore l'algorithme incomplet ci-dessous.

```
pop ← ...
Pour i variant de 1 à ... Faire
  augmentation ← ...
  pop ← pop+...
Fin Pour
Afficher pop
```

1. Compléter puis programmer cet algorithme dans un langage au choix.

2. Modifier puis programmer cet algorithme afin qu'il affiche le nombre d'années nécessaires pour que cette fourmilière dépasse 1 000 fourmis.

7 On considère le programme Python ci-contre.

1. a. Exécuter le programme à la main.

b. Quel est son objectif ?

2. Soit la fonction f définie par $f(x) = -x^2 + 2$ sur $I = [-5; 5]$. Modifier le programme pour qu'il affiche le tableau de valeurs de f pour tous les entiers de I .

```
def f(x):
  return 2*x-3
for x in range(-2, 3):
  print(x,f(x))
```