TP Structure des ordinateurs et applications

Corrigé de la série de TP N°4 – Les instructions de test : Si…Fin-Si & Si…Sinon…Fin-Si

Rappel :

Structures de contrôle conditionnel

Les structures conditionnelles sont utilisées pour contrôler l'exécution d'un bloc d'instructions : elles permettent soit d'exécuter ce bloc, soit de le sauter. Elles servent également à choisir entre l'exécution de deux blocs distincts, en fonction de conditions spécifiques. On distingue principalement deux types de structures conditionnelles :

1. Test alternatif simple

Un test simple comporte un seul bloc d'instructions. En fonction d'une condition (expression logique), on décide si ce bloc doit être exécuté ou non. *Si* la condition est *vraie*, le bloc d'instructions est exécuté ; *sinon*, il est ignoré.

La syntaxe d'un test alternatif simple est la suivante :

Algorithme	Langage C
si <Condition> **alors** <bloc_instructions_si> ; **finsi ;**	**if** (Condition) **{** <bloc_instructions_if> ; **}**

Remarque : ***En langage C,*** un bloc d'instructions est délimité par des accolades ***{ }***. Toutefois, si le bloc ne contient qu'une seule instruction, les accolades deviennent facultatives et peuvent être omises.

2. Test alternatif double

Un test double contient deux blocs d'instructions, et permet de choisir entre l'exécution du premier bloc ou du second. Cette décision repose sur une condition (expression logique ou booléenne) qui peut être évaluée comme vraie ou fausse. ***Si*** la condition est ***vraie***, le premier bloc est exécuté ; ***sinon***, c'est le second bloc qui sera exécuté.

La syntaxe d'un test alternatif double est la suivante :

Algorithme	Langage C
si <Condition> **alors** <bloc_instructions_si> **sinon** <bloc_instructions_sinon> ; **finsi ;**	**if** (Condition) **{** <bloc_instructions_if> ; **}** **else** **{** <bloc_instructions_else> ; **}**

Remarque :

En langage C, il ne faut pas ajouter de point-virgule après la condition dans une instruction ***if*** ou ***else*** ; cela entraînerait une erreur logique, car cela empêcherait le bloc d'instructions de s'exécuter correctement.

Dans l'exemple précédent, les accolades ***{ }*** autour des blocs d'instructions ***if*** et ***else*** peuvent être omises si chaque bloc ne contient qu'une seule instruction. Cela simplifie la syntaxe, mais il est important de veiller à ce que chaque bloc reste clairement défini.

Les structures conditionnelles doubles et imbriquées :

Un test double et imbriqué est similaire à un test double classique, mais il comporte une ou plusieurs conditions supplémentaires dans l'un des blocs d'instructions (le bloc ***Si*** ou le bloc ***Sinon***). Cela signifie qu'au moins un des deux blocs (soit celui de la condition vraie, soit celui de la condition fausse) contient à son tour une condition simple ou une autre structure conditionnelle double. En conséquence, un test double et imbriqué comprend au ***minimum trois blocs*** d'instructions, avec au ***moins deux conditions*** (une condition principale et ***une ou plusieurs conditions imbriquées***).

La syntaxe d'un test alternatif double imbriqué avec trois blocs d'instructions est la suivante :

Algorithme	Langage C

si <Condition> **alors**
  <bloc_instructions_si>
  sinon
  si <Condition> **alors**
  <bloc_instructions_si>
  sinon
  <bloc_instructions_sinon> ;
  finsi
finsi ;
	if (Condition)
	{
	  <bloc_instructions_if> ;
	}
	else
	if (Condition)
	{
	  <bloc_instructions_if> ;
	}
	else
	{
	  <bloc_instructions_else> ;
	}

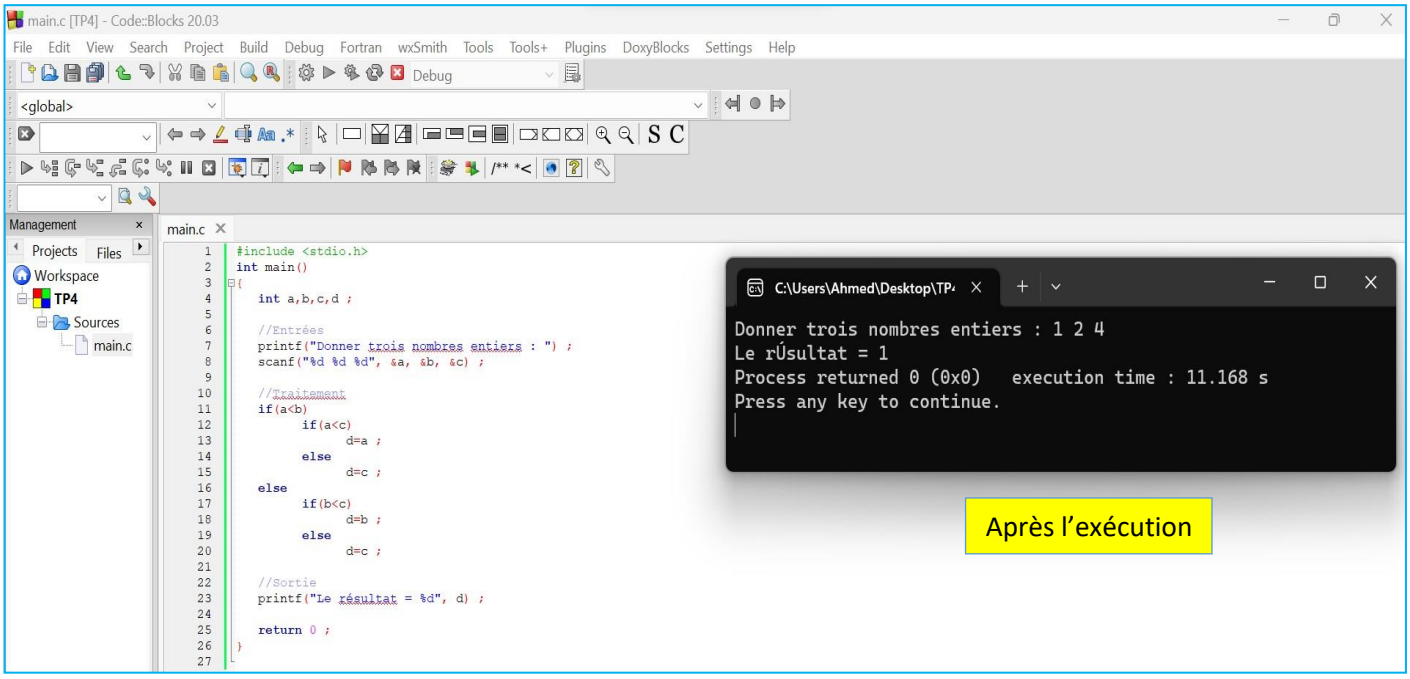
Solution de l'exercice N°01 :

1) Traduction de l'algorithme en un programme en langage C

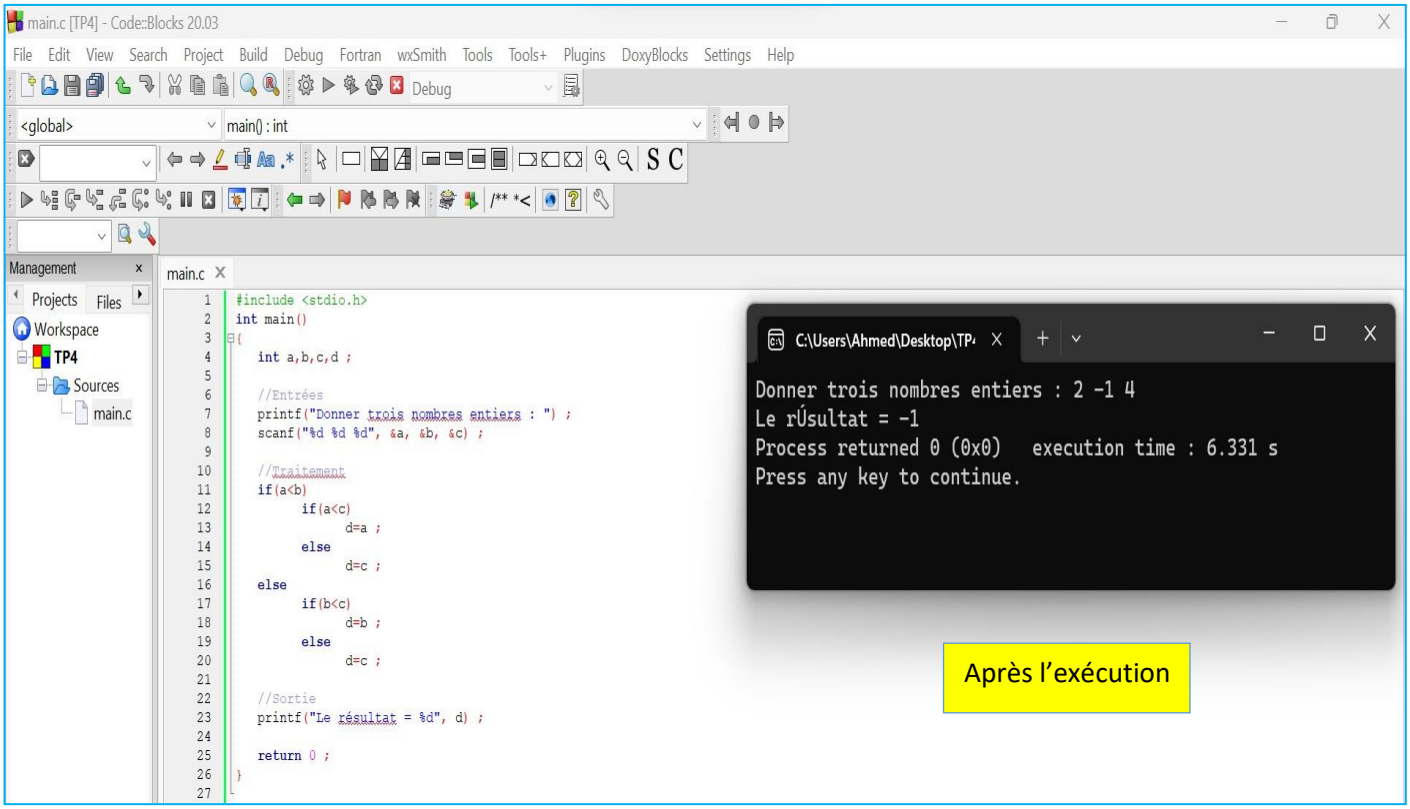
Algorithme	*Programme C*

Algorithme exo1 ;
Variables
a, b, c, d : **entier** ;
Début
//Entrées
Écrire ("Donner trois nombres entiers : ") ;
Lire (a,b,c) ;
//Traitement
Si (a<b) **alors**
Si (a<c) **alors**
d ← a ;
Sinon
d ← c ;
Fin-Si
Sinon
Si (b<c) **alors**
d ← b
Sinon
d ← c ;
Fin-Si
Fin-Si
//Sortie
Écrire ("Le résultat = ", **d**) ;
Fin.
	#include <stdio.h>
	int main()
	{
	int a,b,c,d ;
	//Entrées
	printf("Donner trois nombres entiers : ") ;
	scanf("%d %d %d", &a, &b, &c) ;
	//Traitement
	if(a<b)
	if(a<c)
	d=a ;
	else
	d=c ;
	else
	if(b<c)
	d=b ;
	else
	d=c ;
	//Sortie
	printf("Le résultat = %d", **d**) ;
	return 0 ;
	}

☞ **Compilation et exécution du programme pour les valeurs suivantes :** ***a=1, b=2, c=4***



☞ **Compilation et exécution du programme pour les valeurs suivantes :** ***a=2, b=-1, c=4***



☞ **Déduction de ce que fait l'algorithme/Programme ?**

Le programme donne le plus petit des trois nombres entiers

4-1) Déroulement du programme pour a=1, b=2 et c=4

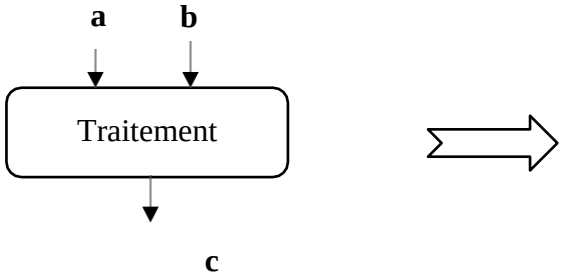
Instructions	**Variables**				**Affichage**
	a	**b**	**c**	**d**	
Écrire ("Donner trois nombres entiers : ") ;	/	/	/	/	Donner trois nombres entiers :
Lire (a,b,c) ;	1	2	4	/	
Si (a<b) **alors** **1 < 2 True** ⇨ On exécute le bloc **Si** **Si** (a<c) **alors** **1 < 4 True** ⇨ On exécute le bloc **Si** d ← a **d ← 1** **Fin-Si**	1	2	4	**1**	
Écrire ("Le résultat = ", d) ;	1	2	4	**1**	Le résultat = **1**

4-3) Déroulement du programme pour a=2, b=-1 et c=4

Instructions	Variables				Affichage
	a	b	c	d	
Écrire ("Donner trois nombres entiers : ") ;	/	/	/	/	Donner trois nombres entiers :
Lire (a,b,c) ;	2	-1	4	/	
Si (a<b) **alors** **2 < -1** **False** ⇨ On n'exécute pas le bloc **Si** (on passe au bloc **Sinon**) **Si** (b<c) **alors** **-1 < 4** **True** ⇨ On n'exécute le bloc **Si** d ← b ; **d ← -1** **Fin-Si**	2	-1	4	**-1**	
Écrire ("Le résultat = ", d) ;	2	-1	4	**-1**	Le résultat = **-1**

Solution de l'exercice N°02:

☞ ***Le schéma entrées, traitement et sorties.***



Entrées :

Lire (a, b) ;

Traitements :

```
Si a=b alors
        c ← 0;
    Sinon
        Si a>b alors
            c ← a*b;
        Sinon
            c ← b+a;
        Fin-Si
Fin-Si
```

Sortie :

Écrire (c) ;

☞ ***Algorithme / Programme en langage C***

Algorithme	Programme en C
Algorithme exo02; ***Variables*** a, b, c : ***entiers*** ; ***Début*** *{Lire les deux nombres entiers}* ***Ecrire*** ("*Entrez deux valeurs de (a et b) :* "); ***Lire***(a,b) ; *{Calculer c selon les conditions}* ***Si*** (a = b) **alors** c ← 0; ***Sinon*** ***Si*** (a > b) c ← a * b; ***Fin-Si*** ***Sinon*** c ← a + b; ***Fin-Si*** *{Afficher le résultat}* ***Ecrire***("La valeur de c est : ", **c**); ***Fin.***	***#include <stdio.h>*** int **main ( )** ***{*** int a, b, c**;** *// Lire les deux nombres entiers* printf("*Entrez deux valeurs de (a et b) : \n*")**;** scanf("**%d**""**%d**" , &a, &b)**;** *// Calculer c selon les conditions* ***if*** (a = = b) ***{*** c = 0**;** ***}*** ***else if*** (a > b) ***{*** c = a * b**;** ***}*** ***else {*** c = a + b**;** ***}*** *// Afficher le résultat* printf("**La valeur de c est : %d\n**", **c**); ***return 0;*** ***}***

☞ ***Compilation et exécution du programme C pour a=5 et b=4 :***

```c
#include <stdio.h>
int main ( )
{
    int a, b, c;
    // Lire les deux nombres entiers
    printf("Entrez deux valeurs de (a et b) : \n");
    scanf("%d""%d"  , &a, &b);

    // Calculer c selon les conditions
    if (a == b) {
        c = 0;
        }
       else if (a > b) {
          c = a * b;
        }
       else {
          c = a + b;
        }
    // Afficher le résultat
    printf("La valeur de c est : %d\n", c);
    return 0;
}
```

```
"C:\Users\Maison\Desktop\Prog-C-SÚrie-1-SE...
Entrez deux valeurs de (a et b) :
5
4
La valeur de c est : 20

Process returned 0 (0x0)   execution time : 4.740 s
Press any key to continue.
```

Solution de l'exercice N°03 :

Algorithme	Programme C

Algorithme Exo03;
Variables
prix_initial, prix_total : réel ;
quantite : Entier ;
Début
{Demander à l'utilisateur d'entrer le prix initial et la quantite}
Écrire("***Entrez le prix initial de l'article (en D.A):*** ");
Lire(prix_initial) ;
Écrire(" **Entrez la quantité achetée :** ");
Lire(quantite) ;
{Calculer le prix total avec remise selon les conditions}
Écrire("Les valeurs triées par ordre croissant sont : ");
Si (prix_initial < 100) ***alors***
Si (quantite < 5) ***alors***
prix_total = prix_initial * quantite * 0.95; // Remise de 5%
Sinon
prix_total = prix_initial * quantite * 0.90; // Remise de 10%
Fin-Si
Fin-Si

Sinon
Si (prix_initial >= 100 && prix_initial <= 500) ***alors***
Si (quantite < 10) ***alors***
prix_total = prix_initial * quantite * 0.85; // Remise de 15%
Sinon
prix_total = prix_initial * quantite * 0.80; // Remise de 20%

Fin-Si
Fin-Si

Sinon
prix_total = prix_initial * quantite * 0.75; // Remise de 25% pour les articles au-dessus de 500 D.A.

Écrire(***Le prix total après remise est :*** " " ***D.A*** " , **prix_total**);

Fin.

```c
#include <stdio.h>

int main ( )
{
  float  prix_initial, prix_total;
  int  quantite;

  // Demander à l'utilisateur d'entrer le prix initial et la quantité
  printf("Entrez le prix initial de l'article (en D.A):\n ");
  scanf("%f", &prix_initial);
  printf("Entrez la quantité achetée : \n");
  scanf("%d" , &quantite);

  // Calculer le prix total avec remise selon les conditions
  if (prix_initial < 100)
{
    if (quantite < 5)
      {
        prix_total = prix_initial * quantite * 0.95;  // Remise de 5%
      }
    else
        prix_total = prix_initial * quantite * 0.90;  // Remise de 10%
 }

 else if  (prix_initial >= 100 && prix_initial <= 500)
{
    if (quantite < 10)
      {
          prix_total = prix_initial * quantite * 0.85;  // Remise de 15%
      }
      else
        prix_total = prix_initial * quantite * 0.80;  // Remise de 20%
}

else
{
  prix_total = prix_initial * quantite * 0.75;  // Remise de 25% pour les articles au-dessus de 500 D.A.
 }

  // Afficher le prix total après remise
  printf("Le prix total après remise est : %.2f D.A\n",
   prix_total);

  return 0;
}
```

☞ ***Compilation et exécution du programme C :***

```c
#include <stdio.h>
 int main ( )
{
    float  prix_initial, prix_total;
    int  quantite;
    // Demander à l'utilisateur d'entrer le prix initial et la quantité
    printf("Entrez le prix initial de l'article (en D.A):\n ");
    scanf("%f", &prix_initial);
    printf("Entrez la quantite achetee : \n");
    scanf("%d" , &quantite);
    // Calculer le prix total avec remise selon les conditions
    if (prix_initial < 100)
{
        if (quantite < 5)
              {
            prix_total = prix_initial * quantite * 0.95;  // Remise de 5%
              }
        else
            prix_total = prix_initial * quantite * 0.90;  // Remise de 10%
 }
 else if  (prix_initial >= 100 && prix_initial <= 500)
{
         if (quantite < 10)
             {
                  prix_total = prix_initial * quantite * 0.85;  // Remise de 15%
             }
             else
            prix_total = prix_initial * quantite * 0.80;  // Remise de 20%
}
else
{
   prix_total = prix_initial * quantite * 0.75;  // Remise de 25% pour les articles au-dessus de 500 D.A.
 }
    // Afficher le prix total après remise
    printf("Le prix total apres remise est : %.2f D.A\n", prix_total);
    return 0;
}
```

```
"C:\Users\Maison\Desktop\Prog-C-SÚrie-1-SEMESTRE...
Entrez le prix initial de l'article (en D.A):
 250
Entrez la quantite achetee :
5
Le prix total apres remise est : 1062.50 D.A

Process returned 0 (0x0)   execution time : 9.910 s
Press any key to continue.
```

Solution de l'exercice N°04 :

Algorithme	Programme C

Algorithme ordre_croissant;
Variables
A, B, C : **entier**;
Début
//Entrées
Écrire("Donner trois valeurs entière A, B et C : ");
Read(A, B, C) ;

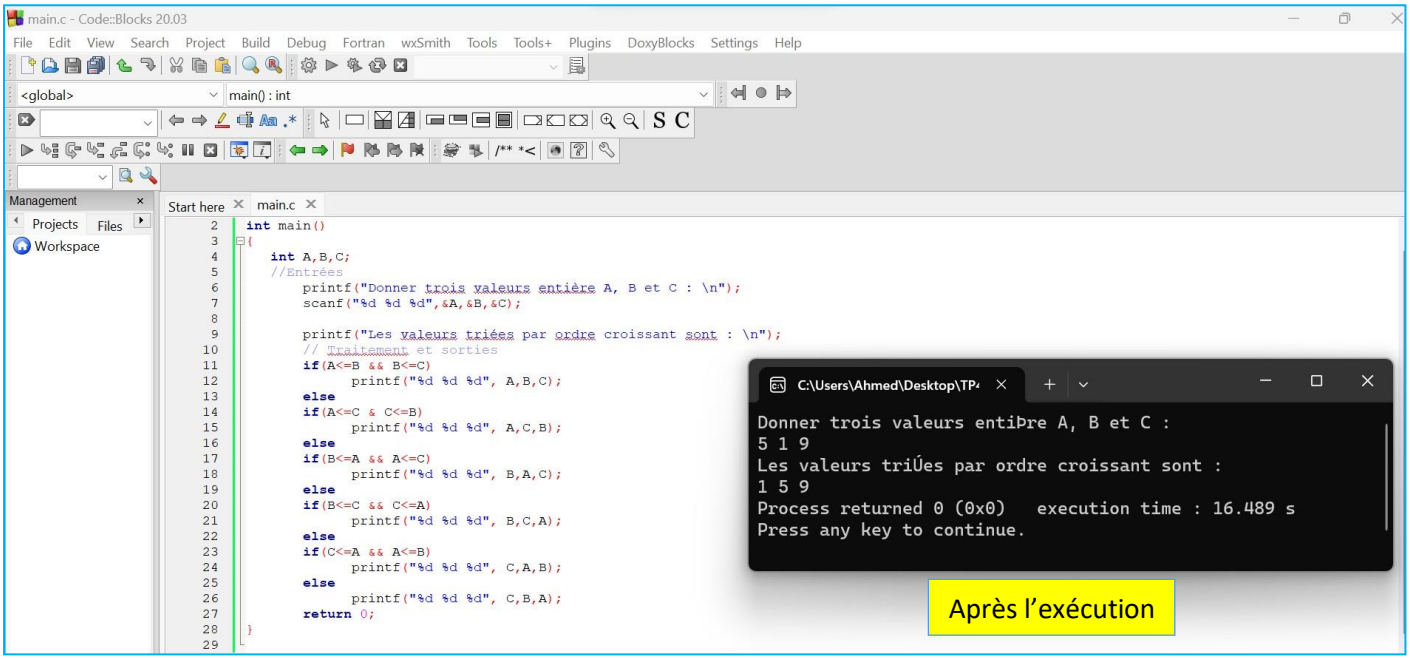
//Traitement & Sorties
Écrire("Les valeurs triées par ordre croissant sont : ");
Si (A <= B **ET** B <= C) alors
Écrire(A, B, C);
Sinon
Si (A <= C **ET** C <= B) **alors**
Écrire (A, C, B) ;
Sinon
Si (B <= A **ET** A <= C) **alors**
Écrire(B, A, C)
Sinon
Si (B <= C **ET** C<=A) **alors**
Écrire(B, C, A)
Sinon
Si (C <= A **ET** A<=B) **alors**
Écrire(C, A, B);
Sinon
Écrire(C, B, A);
Fin-Si ;
Fin-Si ;
Fin-Si ;
Fin-Si ;
Fin-Si ;
Fin.

```c
#include<stdio.h>
int main()
{
  int A,B,C;
  //Entrées
  printf("Donner trois valeurs entière A, B et C : \n");
  scanf("%d %d %d",&A,&B,&C);

  // Traitement et sorties
  printf("Les valeurs triées par ordre croissant sont : \n");
  if(A<=B && B<=C)
      printf("%d %d %d", A,B,C);
  else
  if(A<=C & C<=B)
      printf("%d %d %d", A,C,B);
  else
  if(B<=A && A<=C)
      printf("%d %d %d", B,A,C);
  else
  if(B<=C && C<=A)
      printf("%d %d %d", B,C,A);
  else
  if(C<=A && A<=B)
      printf("%d %d %d", C,A,B);
  else
      printf("%d %d %d", C,B,A);

  return 0;

}
```



N.B : Il existe d'autres solutions qui permettent d'afficher trois valeurs numériques A, B et C avec ordre croissant.