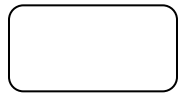
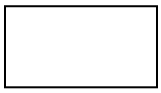
Chapitre 3 : Les structures conditionnelles

1. Introduction :

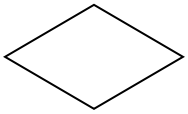
- Les ***structures de contrôle*** encore appelées ***instructions (actions) structurées*.**
  Elles permettent d'exprimer la façon de l'enchaînent d'exécution des instructions d'un algorithme.
- Il existe trois structures fondamentales :
  1) Actions séquentielles,
  2) Actions conditionnelles,
  3) Actions de répétition.
- Pour décrire ces structures on utilise une notation textuelle (***algorithme***) et une notation graphique (***organigramme***).
- Dans un organigramme on utilise les symboles suivants :


Représente le début et la Fin de l'organigramme


Entrées / Sorties : Lecture des données et écriture des résultats.


Représente les Actions (traitement)


Représente l'ordre d'exécution des opérations(Enchaînement)

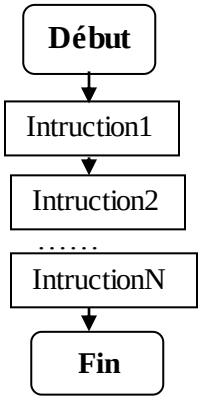

Représenter les conditions (Tests et décision)

2. Actions séquentielles :

- Les Actions séquentielles se présentent sous forme d'une suite ordonnée d'instructions regroupée en un bloc.

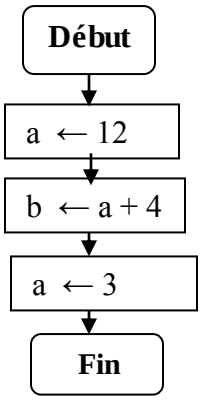
Syntaxe :

Début
Intruction1 ;
Intruction2 ;
……
Instruction N ;
Fin.



Exemple :

Algorithme exp_sequ
a, b : entier;
Début
a← 12 ; (Instruction 1)
b←a + 4 ; (Instruction 2)
a ← 3 ; (Instruction 3)
Fin.



Trace d'exécution :

	a	b
début	**?**	**?**
a← 12 ;	**12**	**?**
b←a + 4 ;	**12**	**16**
a ← 3 ;	**3**	**16**
Fin.	**3**	**16**

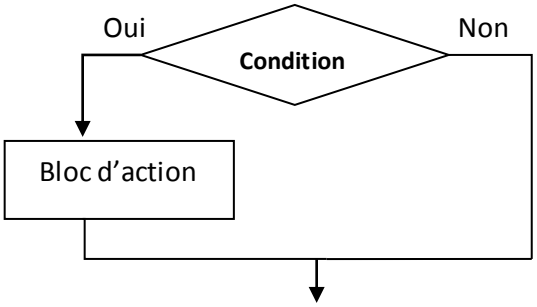
3. Action conditionnelle :

3.1. Action conditionnelle simple :

- Elle est composée de deux parties : ***condition*** et ***action***.
  - ✓ La partie ***(condition)*** décrit un état qui peut être vrai ou faux (expression de type Booléen).
  - ✓ La partie ***< Bloc d'actions >*** représente un morceau d'un algorithme (une ou plusieurs instructions).

Syntaxe :

…….
Si (condition) **alors**
< Bloc d'actions (instructions)>
FinSi
……



Exécution de l'action conditionnelle :

- Si la condition est ***vérifiée (vrai)***, les instructions du < Bloc d'actions > sont exécutées et on continue l'exécution des actions (instructions) situées après le Finsi.
- Si la condition n'est ***pas vérifiée (faux)***, la partie < Bloc d'actions > à l'intérieur du Si n'est pas exécutée et on poursuit l'exécution de l'algorithme directement à partir de l'instruction qui suit le FinSi.

Exemple : Ecrire un algorithme qui permet de lire un nombre réel identifier par 'Nbr', puis donne sa valeur absolue.

Algorithme val_abs

Nbr : réel ;

Début

Lire(Nbr) ;

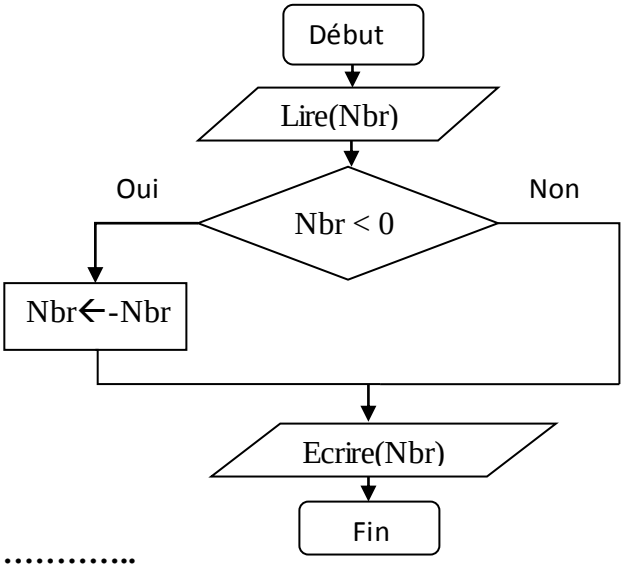
Si (Nbr< 0) **alors**

Nbr← - Nbr ;

FinSi

Ecrire (Ndr) ;

Fin.



Syntaxe en langage C++ :

…….	……..
Si (condition) **alors**	**if** (condition) **{**
< Bloc d'actions (instructions)>	< Bloc d'actions (instructions)>
FinSi	**}**

Remarque :

- la condition est une ***expression*** de type ***Booléen*** donc elle doit comporter au moins un opérateur de comparaison (<, >, =, ≠,…etc.) ou une ***variable Booléen***.

3.2. Action alternative :

Syntaxe :

……

Si (condition) **alors**

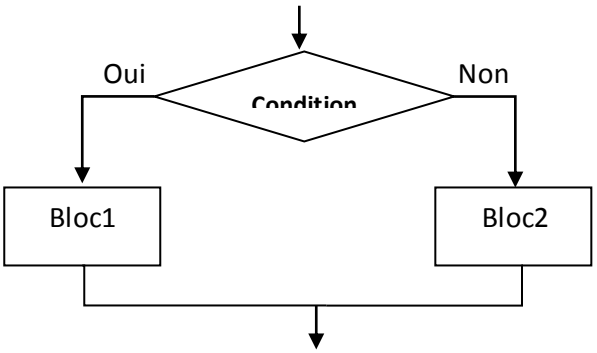
< Bloc d'actions**1** (instructions) >

Sinon

< Bloc d'actions**2** (instructions) >

FinSi

……



Exemple : Ecrire un algorithme qui permet de lire deux nombres réels, puis détermine le plus grand entre eux ?

Algorithme PlusGrNbr

x, y, Max : réel ;

Début

Lire(x) ;

Lire(y) ;

Si (x > y) **alors**

Max← x ;

Sinon

Max← y ;

FinSi

Ecrire ('Le plus grand nombre est :', Max);

Fin.

Syntaxe en langage C++

……	……
Si (condition) **alors**	**if** (condition) **{**
< Bloc d'actions**1** (instructions) >	< Bloc d'actions**1** (instructions) >
	}
Sinon	**else**
	{
< Bloc d'actions**2** (instructions) >	< Bloc d'actions**2** (instructions) >
FinSi	**}**
……	…….

Remarque:

- On peut voir une structure de contrôle en entier comme une seule action (< Bloc d'actions >) donc il peut y a avoir plusieurs ***structures de contrôle imbriquées***.

```
Si (condition1) alors
        Si (condition2) alors
        < Bloc d'actions1 (instructions) >
        Sinon
        < Bloc d'actions3 (instructions) >
        FinSi
Sinon
        < Bloc d'actions2 (instructions) >
    FinSi
```

Exemple : Ecrire un algorithme qui permet de lire deux nombres réels, puis détermine le plus grand entre eux ?

```
Algorithme plus_G
        x, y, Max : réel ;
Début
  Lire(x, y) ;
  Si (x = y) alors
Ecrire ('Le deux sont égaux');
Sinon   // x ≠ y
       Si (x > y) alors
        Max ← x ;
        Ecrire ('Le plus grand nombre est :', Max);
        Sinon  // x < y
        Max ← y ;
        Ecrire ('Le plus grand nombre est :', Max);
        FinSi
FinSi
Fin.
```

3.3. Action de choix multiple :

- Elle permet de distinguer plusieurs cas selon les ***valeurs*** d'une ***expression***.
- Le « **si** » permet de distinguer deux cas alors que le « **Selon cas** » permet de distinguer un grand nombre de cas.

Syntaxe :

Selon cas (expression**)**

Cas 1 : <bloc actions 1>

Cas 2 : <bloc actions 2>

……..

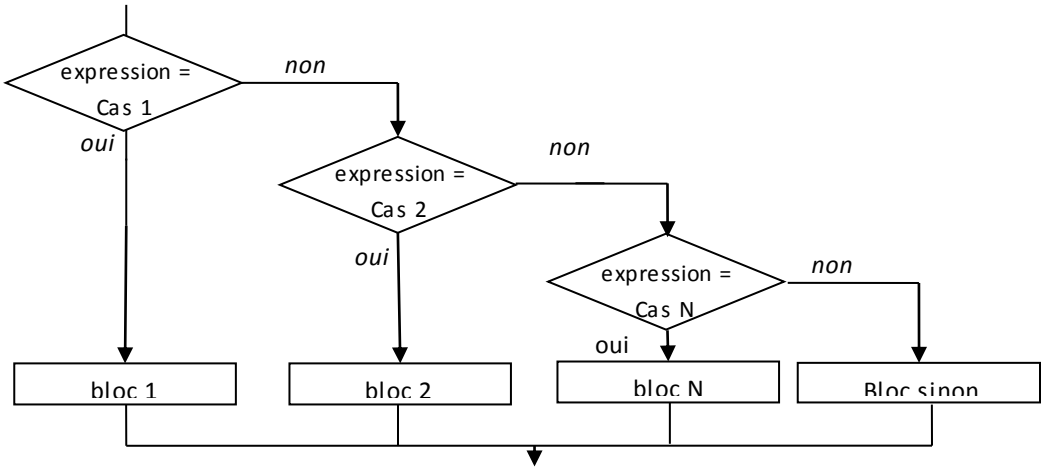
Cas N : <bloc actions N>

Sinon // Le sinon est optionnel

<bloc action Sinon>

Fin cas

L'organigramme de la structure « Selon cas » est le suivant :



Syntaxe en langage C++

Selon cas (expression**)**	**Switch (**expression**) {**
Cas 1 : <bloc actions 1>	**case** val1 **: {** <bloc actions 1> **break ; }**
Cas 2 : <bloc actions 2>	**case** val2 **: {** <bloc actions 2> **break ;}**
……..	……..
Cas N : <bloc actions N>	**case** valN **: {** <bloc actions N> **break ;}**
Sinon // Le sinon est optionnel	**default**
<bloc action Sinon>	**{** <bloc action default> **break ; }**
Fin cas	**}**

Remarque : l'instruction « **break ;** » est nécessaire pour sortir de la structure selon cas une fois on exécute le bloc d'action correspondant ; pour ne pas tester les autres cas, mais elle n'est pas obligatoire.

Exemple : Ecrire l'algorithme qui permet de lire un chiffre (0, 1, 2, 3, 4) puis donne son nom (zéro, un, deux, trois, quatre) ?

Action alternative imbriquée	**Action à choix multiple**
Algorithme Nom_chif n : entier ; **Début** Ecrire ('donnez votre chiffre entre 0 et 4 : ') ; Lire (n) ; Si (n=0) Alors Ecrire ('Zéro') ; Sinon Si (n=1) Alors Ecrire ('Un') ; Sinon Si (n=2) Alors Ecrire ('Deux') ; Sinon Si (n=3) Alors Ecrire ('Trois') ; Sinon Si (n=4) Alors Ecrire ('Quatre') ; Sinon Ecrire ('erreur de la saisie ') ; Fin si Fin si Fin si Fin si Fin si **Fin.**	**Algorithme** Nom_chif n : entier ; **Début** Ecrire ('donnez votre chiffre entre 0 et 4 : ') ; Lire (n) ; **Selon Cas** (n) 0 : Ecrire (' Zéro') ; 1 : Ecrire ('Un') ; 2 : Ecrire ('Deux') ; 3 : Ecrire ('Trois') ; 4 : Ecrire ('Quatre') ; **Sinon** Ecrire ('erreur de la saisie') ; **Fin cas** **Fin.**