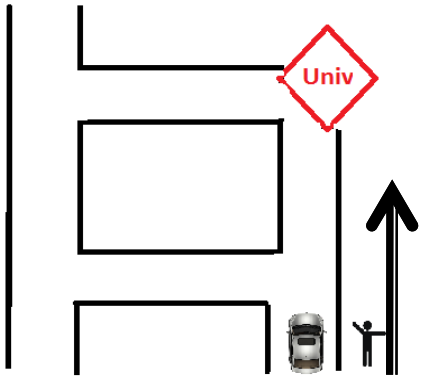
Initiation à l'Algorithmique

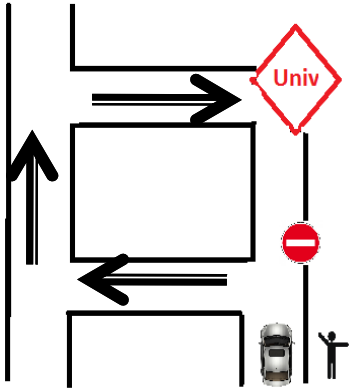
Algorithmique

ENS KATEB

Les structures Conditionnelles : (Si … alors … sinon …FinSi)



Voiture : je veux me rendre à l'université
Piéton : allez tout droit



Voiture : je veux me rendre à l'université
Piéton : prenez la première sortie à gauche puis la première à droite ensuite la première à droite

Cas où le piéton ne sait pas s'il y a un panneau de sens interdit ou pas :

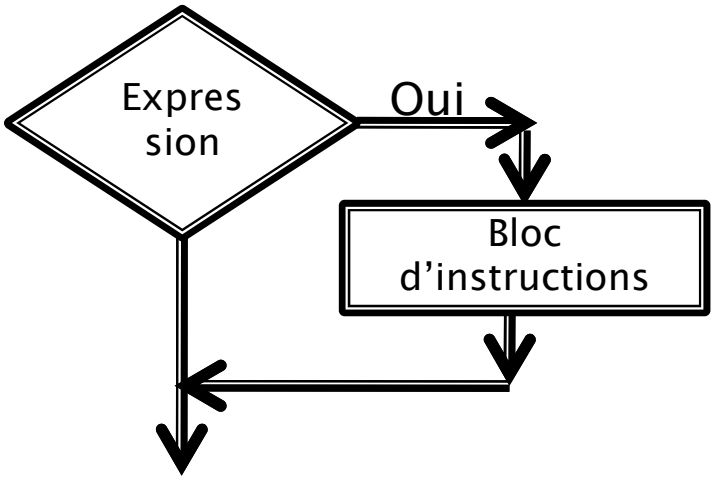
Voiture : je veux me rendre à l'université
Piéton : Si il n'y a pas sens interdit **alors** allez tout droit **sinon** prenez la première sortie à gauche puis la première à droite ensuite la première à droite

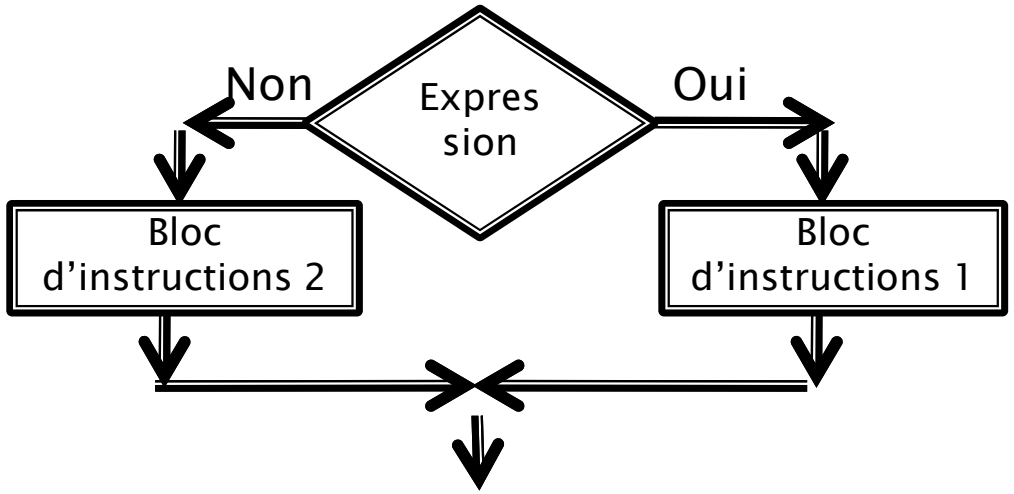
Syntaxe de test

Si (expression) alors
    Bloc d'instructions;
Finsi;

ou

Si (expression) alors
    Bloc d'instructions 1;
Sinon
    Bloc d'instructions 2;
Finsi;





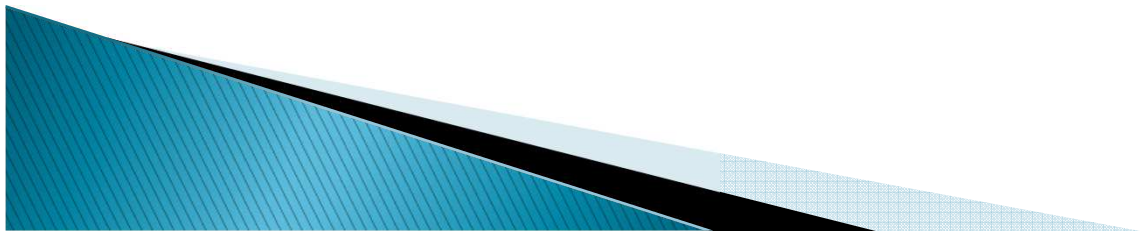
Expression est de type booléen

Pour cela on utilise les opérateurs de relation (=, <>, <, >, <=, >=)

Exemple

Écrire un algorithme qui demande à l'utilisateur d'entrer un nombre entier puis indique si ce nombre est positif ou négatif (considérons que 0 est positif)

```
Algo Signe
Var X : entier;
Début
  Écrire (''Donner un nombre:'');
  Lire(X);
  Si (X>=0) alors
     Écrire (''le nombre saisi est positif'');
  Sinon
     Écrire (''le nombre saisi est négatif'');
  FinSi;
Fin.
```



Conditions Composées

- On peut avoir des conditions composées :
- Si le ciel est nuageux et la température est inférieure à 20 alors nous somme en hiver
- Si ((cond1 et cond2) ou (cond3 et cond4)) alors ……
- ……

Écrire un algorithme qui demande à l'utilisateur d'entrer deux nombres entiers puis indique si leur produit est positif ou négatif (considérons que 0 est positif) sans effectuer le produit

```
Algo Signe_prod
Var X,Y : entier;
Début
  Écrire (''Donner deux nombres:'');
  Lire(X,Y);
   Si (((X>=0) et (Y>=0)) ou ((X<0) et (Y<0)))
   alors
       Écrire (''le produit est positif'');
   Sinon
       Écrire (''le produit est négatif'');
   FinSi;
Fin.
```

Conditions Imbriquées

Exemple :

```
Si cond1 alors
    Si cond 2 alors
        Si Cond 3 alors
            Inst 1;
        Sinon
            Inst 2;
        FinSi
    FinSi
FinSi
```

Écrire un algorithme qui demande à l'utilisateur d'entrer un nombre Puis, s'il est positif, il indique si ce nombre est pair ou Impair sinon (le nombre est négatif) il affiche un message d'erreur.

```
Algo Signe
Var X : entier;
Début
  Écrire (''Donner un nombres:'');
  Lire(X);
  Si (X>=0) alors
     Si ((X mod 2) = 0) alors
        Écrire (''le nombre est pair'');
     Sinon
        Écrire (''le nombre est impair'');
     FinSi;
  Sinon
     Écrire (''Erreur : Vous avez tapez un
                nombre négatif'');
  FinSi;
Fin.
```

Exemple

Ecrire un algorithme qui demande à l'utilisateur d'enter un âge (en ans) et l'informe ensuite de sa catégorie :
Entre 00 et 15 : Enfant
Entre 16 et 33 : Jeune
Entre 34 et 60 : pépère
Plus de 61 : Vioc
Moins de 00 : Erreur

```
Algo Catégorie
Var age : entier ;
Début
  Ecrire (''Entrer l'âge :'') ;
  Lire (age) ;
  Si (age >= 0) alors
     Si (age <=15) alors
        Ecrire (''c'est un enfant'') ;
     Sinon
        Si ((age >=16) et (age <=33)) alors
           Ecrire (''c'est un jeune'') ;
        Sinon
           Si ((age >=34) et (age <=60)) alors
              Ecrire (''c'est un pépère'') ;
           Sinon
              Ecrire (''c'est un vioc'') ;
           FinSi;
        FinSi;
     FinSi;
  Sinon
      Ecrire (''Erreur : le nombre saisi est négatif'') ;
  FinSi;
Fin.
```

En C++

Syntaxe :

```
Si (Condition) Alors
 Bloc d'Instructions;
FinSi
```

En C++:

```
if (Condition) Instruction1;
```

```
if (Condition) {Bloc d'Instructions;}
```

```
Si (Condition) Alors
 Bloc d'Instructions1;
Sinon
 Bloc d'Instructions2;
FinSi
```

```
if (Condition) Instruction1;
else  Instruction2;
```

```
if (Condition) {Bloc d'Instructions1;}
else {Bloc d'Instructions2;}
```

Structures Alternatives : Selon

Exemple : écrire un algorithme qui demande à l'utilisateur de taper un nombre entre 1 et 12 puis affiche le mois correspondant : ex : pour 1 afficher ''janvier''

- Trop d'imbrications
- Difficile à implémenter
- Solution :
  utiliser les structures alternatives : Selon

```
Algo mois
Var N : entier ;
Début
  Ecrire (''Entrer le nombre entre 1 et 12 :'') ;
  Lire (N) ;
  Si (N=1)
     Ecrire (''Janvier'') ;
  Sinon
     Si (N=2) alors
        Ecrire (''Fevrier'') ;
     Sinon
        ……
           ……
              ……
                 Si (N=12) Alors
                    Ecrire (''Fevrier'') ;
Fin.
```

Structures Alternatives : Selon

- Syntaxe :

```
Selon (VE) vaut
 Val1:Bloc1;
 Val2:Bloc2;
      ...
 ValN:BlocN;
 autre:BlocM;
FinSelon
```

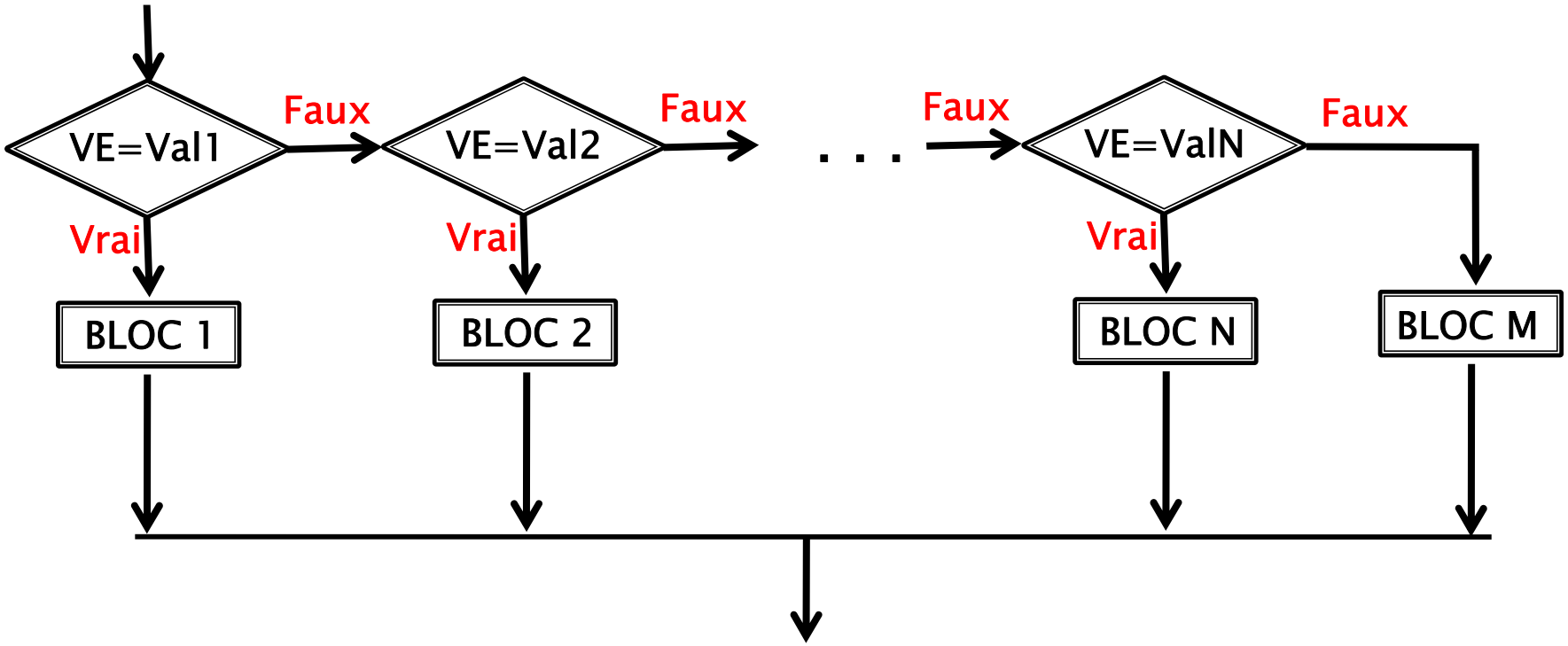
Exemple précédent :

```
Algo mois
Var N : entier ;
Début
  Ecrire (''Entrer un nombre entre 1 et 12 :'') ;
  Lire (N) ;
  Selon (N) vaut
     1  :  Ecrire (''Janvier'') ;
     2  :  Ecrire (''Février'') ;
     3  :  Ecrire (''Mars'') ;
           ...
     12  :  Ecrire (''Décembre'') ;
     autre  :  Ecrire (''Valeur incorrecte'') ;
  FinSelon;
Fin.
```

VE : Variable ou Expression.
Si VE n'a aucune valeur parmi (Val1, Val2,……., ValN) , le bloc autre sera exécuté.

Structures Alternatives : Selon

- Organigramme :



Structures Alternatives : Selon

- En C++ :

Syntaxe	En C++
Selon (*VE*) **vaut** *Val1:Bloc1;* *Val2:Bloc2;* ... *ValN:BlocN;* **autre:***BlocM;* **FinSelon**	**switch**(*VE*) **{** **case** *Val1:Bloc1;* **break;** **case** *Val2:Bloc2;* **break;** ... **case** *ValN:BlocN;* **break;** **default:***BlocM;* **}**

Structures Alternatives : Selon

- Exemple :

Exemple : écrire un algorithme qui demande à l'utilisateur de taper deux nombres X et Y puis lui demande d'enter un choix sous forme de caractère :
'A' : affiche le résultat de X+Y
'S' : affiche le résultat de X-Y
'M' : affiche le résultat X*Y
'D' :affiche le résultat X/Y
Si l'utilisateur tape autre chose il lui envoi un message d'erreur.

```
Algo arithmétique
Var
    X,Y : Réel ;
    C : caractère;
Début
  Ecrire (''Entrer le 1ere nombre:'') ; Lire (X);
  Ecrire (''Entrer le 2eme nombre:'') ; Lire (Y);
  Ecrire (''Entrer Votre choix (A, S, M ou
            D):'') ; Lire (C);
  Selon (C) vaut
    'A' : écrire (''Addition :'', X+Y);
    'S' : écrire (''Soustraction :'', X-Y);
    'M' : écrire (''Multiplication :'', X*Y);
    'D' : si (y<>0) alors
            écrire (''Division :'', X/Y);
          sinon
            écrire (''Erreur : Division par zero'');
          FinSi;
     autre : écrire (''Erreur : choix incorrect'');
  FinSelon;
Fin.
```