Introduction à l'algorithmique

SNIR

« Un langage de programmation est une convention pour donner des ordres à un ordinateur. Ce n'est pas censé être obscur, bizarre et plein de pièges subtils. Ça, ce sont les caractéristiques de la magie. »

Dave Small

- Objectif : obtenir de la « machine » qu'elle effectue un travail à notre place
- Problème : expliquer à la « machine » comment elle doit s'y prendre

Introduction à l'algorithmique

SNIR

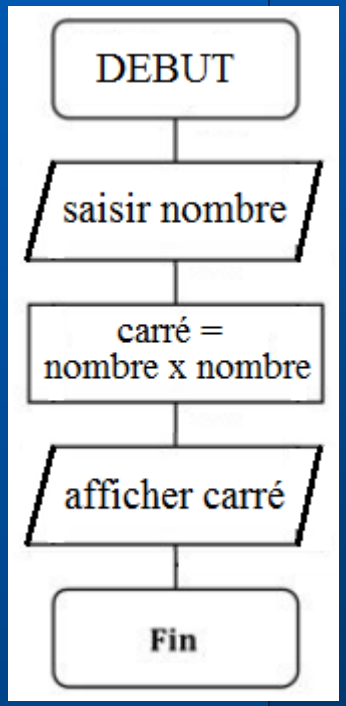
Algorithme et organigramme

- Un algorithme est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème.
- Un organigramme est une représentation d'une programmation sous forme d'un schéma.
- Un programme est une implémentation d'un algorithme ou d'un organigramme.

Introduction à l'algorithmique

SNIR

I) Séquence linéaire algorigramme et algorithme



Début
Saisie
Traitement
Affichage
Fin

```
Algorithme ElèveAuCarré

variable unNombre, sonCarré : entier ;
début
        unNombre := saisir("Quel nombre voulez-vous élever au carré ?") ;
        sonCarré := unNombre × unNombre ;
        afficher("Le carré est : ", sonCarré) ;
fin
```

Introduction à l'algorithmique

SNIR

Transcription en C++

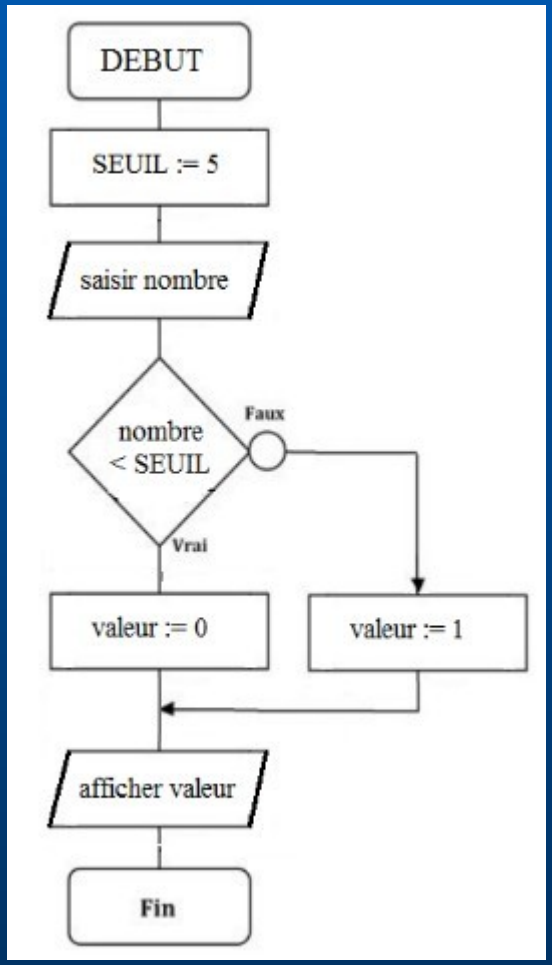
```
#include <iostream>

using namespace std;

int main()
{
  // declaration
   int nombre, carre;

  // saisie
   cout << "tapez un nombre : ";
   cin >> nombre;

  // traitement
   carre = nombre * nombre;

  // affichage
   cout << "son carre est " << carre << endl;

   return 0;
}
```

Introduction à l'algorithmique

SNIR

II) Structure conditionnelle

```
Algorithme ToutOuRien
{ affiche 0 si une valeur saisie est inférieure
à un seuil donné sinon affiche 1 }
constante (SEUIL : entier) := 5 ; { seuil à 5 V }
variable    val : réel ;              { valeur analogique }
début
            val := saisir("Donnez un nombre :") ;
            si val < SEUIL
            alors
                        val := 0 ;
            sinon
                        val := 1 ;
            afficher("La valeur finale est : ", val) ;
fin
```



Introduction à l'algorithmique

SNIR

Transcription en C++

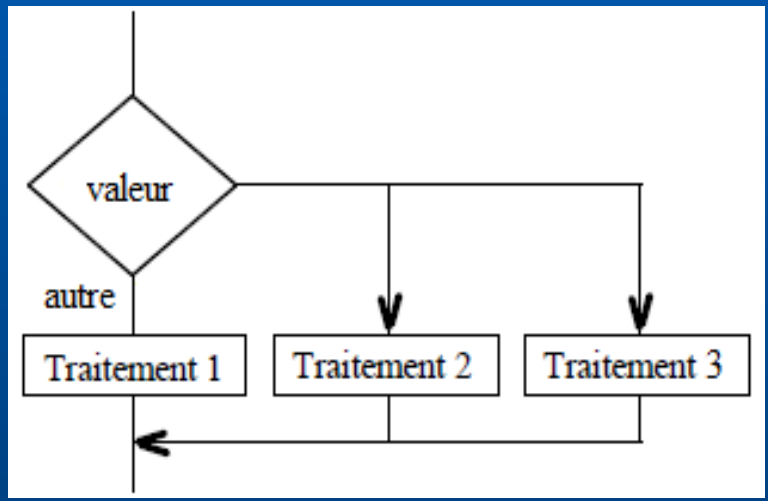
```
#include <iostream>

using namespace std;

int main()
{
   const int seuil(5);
   float val;

   cout << "Donnez un nombre : ";
   cin >> val;

   if ( val <  static_cast<float>(seuil) )
      val = 0;
   else
      val = 1;

   cout << "La valeur finale est : " << static_cast<int>(val);

   return 0;
}
```

Introduction à l'algorithmique

SNIR

Sélection conditionnelle

```
Selon <identificateur> Faire
        (Liste de) valeur :
                Traitement() ;
                FinFaire

        Sinon :
                Traitement_par_defaut() ;
finselon
```



Introduction à l'algorithmique

SNIR

En langage C/C++

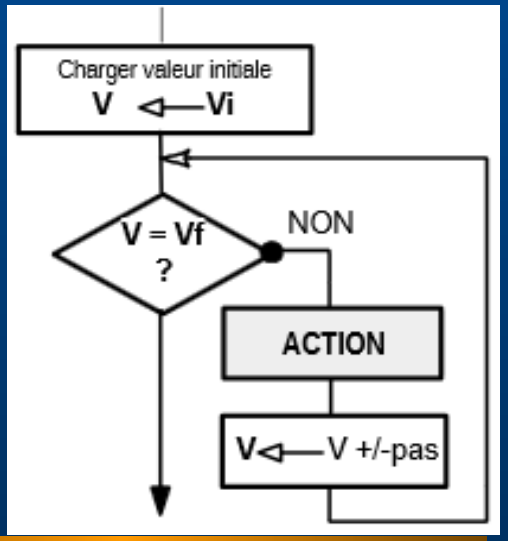
```
switch ( value )
{
   case 2 :
      Traitement2();
      Break ;
   case 3 :
      Traitement3();
      break ;
   default :
      Traitement1();
      break ;
}
```

Introduction à l'algorithmique

SNIR

Les instructions itératives

pour <variable> de <valeur_initiale> à <valeur_finale>
par pas de <n>
Faire
Action(s)
FinFaire



Introduction à l'algorithmique

SNIR

En langage C++

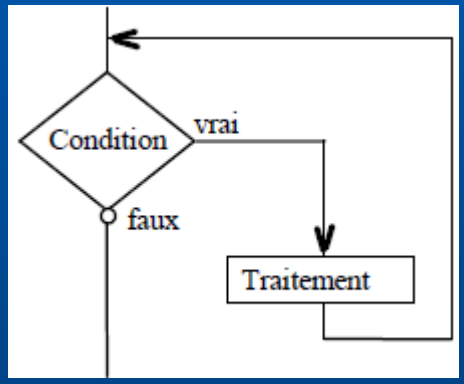
```
// le nombre d'itérations est connu
for ( int i(Vi) ; i != Vf ; i += pas )
{
   Action();
}
// en sortie de boucle, i = Vf
```

Introduction à l'algorithmique

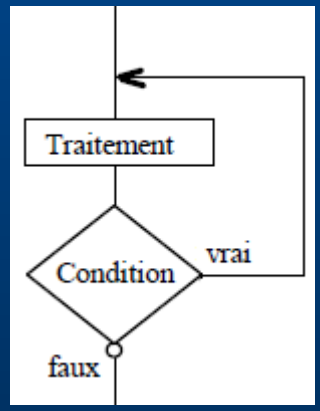
SNIR

Les instructions itératives

1. tanque <expression logique (vraie)> faire
début
Instructions ;
Fin



2. répéter
Instructions ;
tanque <expression logique (vraie)> ;



Introduction à l'algorithmique

SNIR

En langage C/C++

```
while ( condition )
{
   Traitement();
}
```

1

```
do
{
   Traitement();
} while ( condition ) ;
```

2