Introduction à l'algorithmique

Informatique et Science du Numérique

« 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

Informatique et Science du Numérique

Informatique

Science du traitement rationnel de l'information
par des machines automatiques

Philippe Dreyfus, 1962

Introduction à l'algorithmique

Informatique et Science du Numérique

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

Informatique et Science du Numérique

Exemples d'algorithmes

Briques de LEGO → suite de dessins → Camion de pompiers

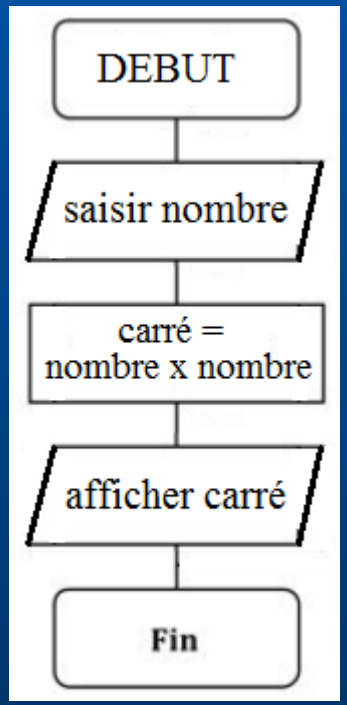
Meuble en kit → notice de montage → Cuisine équipée

Farine, œufs, ... → recette → gâteau

Introduction à l'algorithmique

Informatique et Science du Numérique

Séquence linéaire



Début
Saisie
Traitement
Affichage
Fin

Introduction à l'algorithmique

Informatique et Science du Numérique

Un premier algorithme

```
Algorithme ElèveAuCarré
{ Cet algorithme calcule le carré du nombre que lui fournit l'utilisateur }
variable unNombre, sonCarré : entier ;                { déclarations des variables }
début
        { préparation du traitement }
        unNombre := saisir("Quel nombre voulez-vous élever au carré ?") ;
        { traitement : calcul du carré }
        sonCarré := unNombre × unNombre ;
        { présentation du résultat }
        afficher("Le carré est : ", sonCarré) ;
fin
```

Introduction à l'algorithmique

Informatique et Science du Numérique

```c
#include <stdio.h>

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

   // saisie
   printf("tapez un nombre");
   scanf("%d", &nbr);

   // traitement
   carre = nombre * nombre;

   // affichage
   printf("son carre est %d", carre);

   return 0;
}
```

```cpp
#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;
}
```

```python
# saisie
nombre = input("tapez un nombre : ")

# traitement
carre = nombre * nombre

# affichage
print(carre)
```

Introduction à l'algorithmique

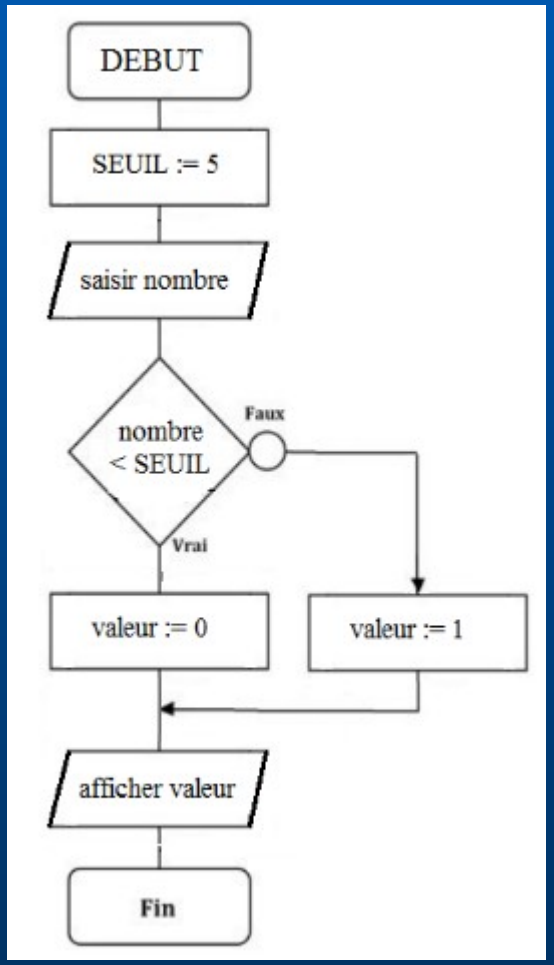
Informatique et Science du Numérique

Un deuxième algorithme

```
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

Informatique et Science du Numérique



Introduction à l'algorithmique

Informatique et Science du Numérique

```c
#include <stdio.h>

int main()
{
   const int SEUIL = 5;
   float val;

   printf("Donnez un nombre : ");
   scanf("%f", &val);

   if ( val < (float) SEUIL )
      val = 0;
   else
      val = 1;

   printf("La valeur finale est : %d", (int) val);

   return 0;
}
```

```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;
}
```

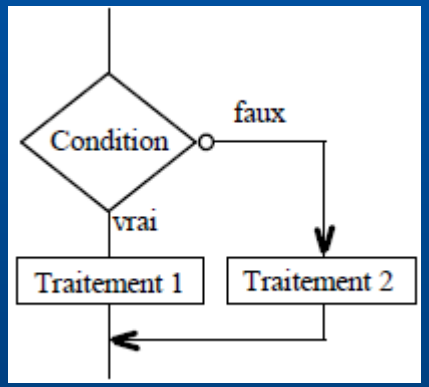
```python
# declaration
SEUIL = 5

# saisie
val = input("Donnez un nombre : ")

# traitement
if val < SEUIL:
      val = 0
else:
      val = 1

# affichage
print("La valeur finale est : ", val)
```

Introduction à l'algorithmique

Informatique et Science du Numérique

L'instruction conditionnelle

```
si <expression logique (vraie)>
alors
        début
                Instructions ;
        fin
sinon
        début
                Instructions ;
        fin
```



Introduction à l'algorithmique

Informatique et Science du Numérique

En langage C/C++

```
if ( condition )
{
        traitement1();
}
else
{
        traitement2();
}
```

En langage Python

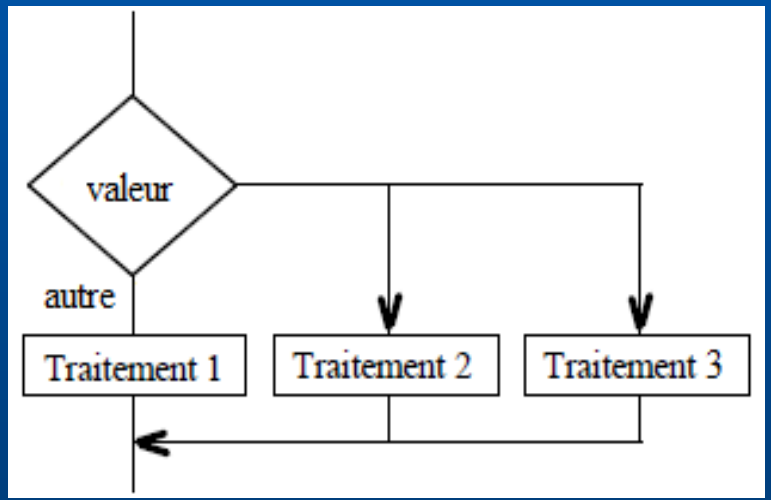
```
if condition :
        traitement1()
else :
        traitement2()
```

Introduction à l'algorithmique

Informatique et Science du Numérique

Sélection conditionnelle

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

        Sinon :
                Traitement_par_defaut() ;
                FinFaire
Fin
```



Introduction à l'algorithmique

Informatique et Science du Numérique

En langage C/C++

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

En langage Python

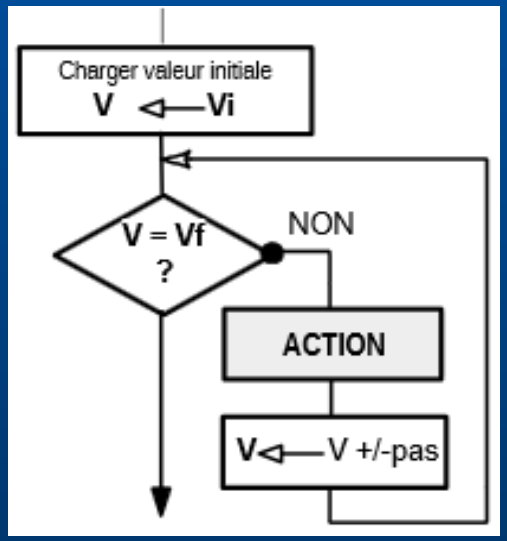
```
if value == 2 :
        Traitement2()
elif value == 3 :
        traitement3()
else :
        traitement1()
```

Introduction à l'algorithmique

Informatique et Science du Numérique

Les instructions itératives

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



Introduction à l'algorithmique

Informatique et Science du Numérique

En langage C

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

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

Informatique et Science du Numérique

En langage Python

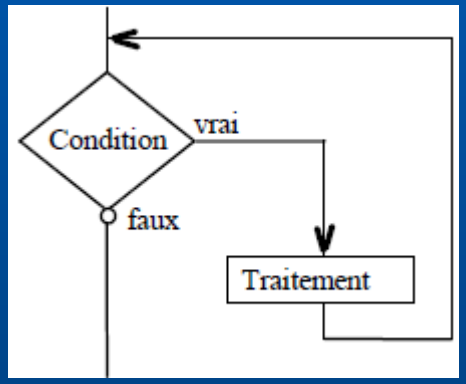
```
// le nombre d'itérations est connu
for i in range(Vi, Vf) :
    Action()
```

Introduction à l'algorithmique

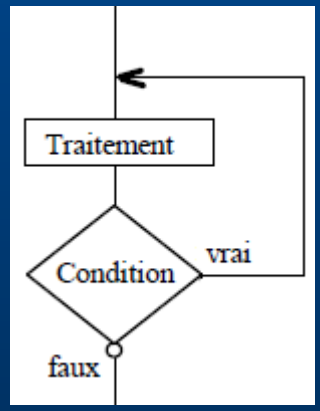
Informatique et Science du Numérique

Les instructions itératives

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



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



Introduction à l'algorithmique

Informatique et Science du Numérique

En langage C/C++

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

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

En langage Python

```
while condition :
   Traitement()
```

```
continuation = True
while continuation :
   Traitement()
   if not condition :
      continuation = False
```

Introduction à l'algorithmique

Informatique et Science du Numérique

Exemple 1 : le distributeur de boisson

Un distributeur propose de 2 types de boissons : eau et soda.
Le stock initial de chaque boisson est égal à 20.
Quand les stocks sont vides, le système doit avertir la maintenance et se mettre Hors Service.
Sinon, le système doit demander la boisson désirée.

- Le bouton 1 sélectionne une bouteille d'eau
- Le bouton 2 sélectionne une canette de soda

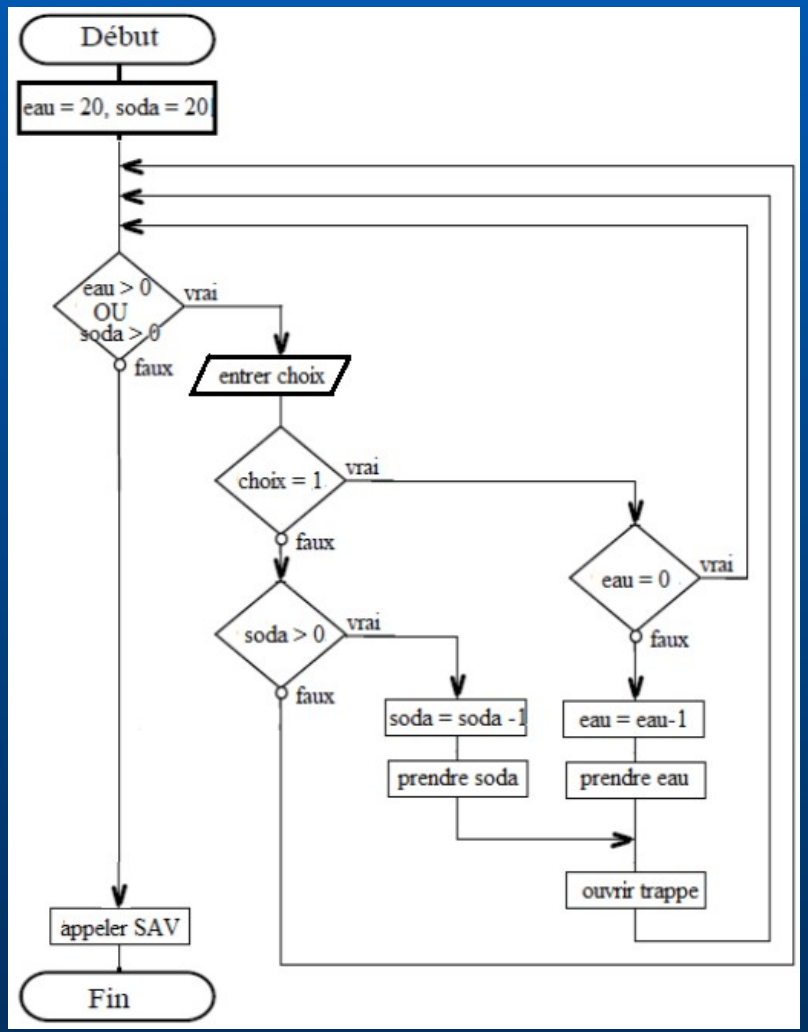
Une fois la sélection faite, si le stock de la boisson sélectionnée n'est pas vide, le système met à jour le stock, sélectionne la boisson demandée et ouvre la trappe d'accès à la boisson.

Introduction à l'algorithmique

Informatique et Science du Numérique

Notions abordées

- Opérateur logique
- Conditions imbriquées

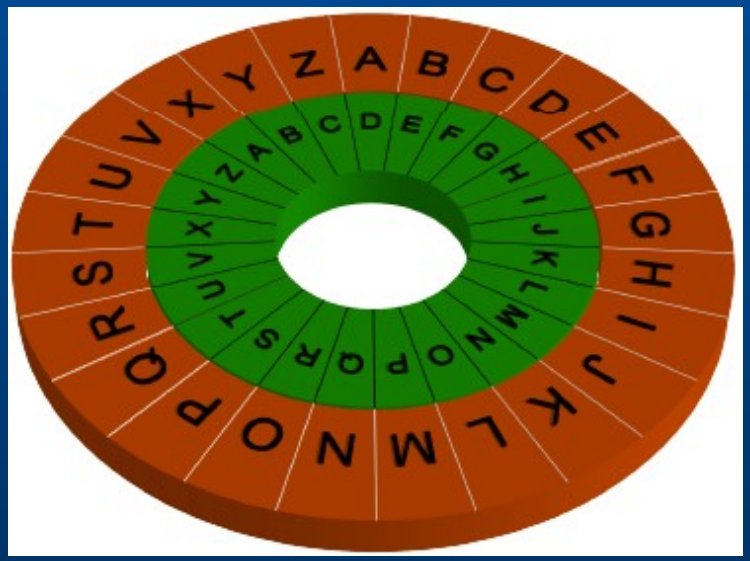


Introduction à l'algorithmique

Informatique et Science du Numérique

Exemple 2 : le chiffre de Caesar*

Jules César, dans ses correspondances secrètes, codait le texte en remplaçant chaque lettre du texte clair original par une lettre à distance fixe, toujours du même côté, dans l'ordre de l'alphabet.



* http://fr.wikipedia.org/wiki/Chiffrement_par_décalage

Introduction à l'algorithmique

Informatique et Science du Numérique

Exemple 3 : le mot de passe

Réaliser l'algorigramme d'un programme qui demande à un utilisateur de définir un mot de passe.

- Le mot de passe ne doit pas être inférieur à 5 caractères.
- Le mot de passe de doit pas dépasser 10 caractères.
- Tant que le mot de passe est incorrect, on doit demander le mot de passe.
- Si le mot de passe est correct, on fait appel à un sous programme de chiffrement*, puis on enregistre le mot de passe chiffré dans un fichier.

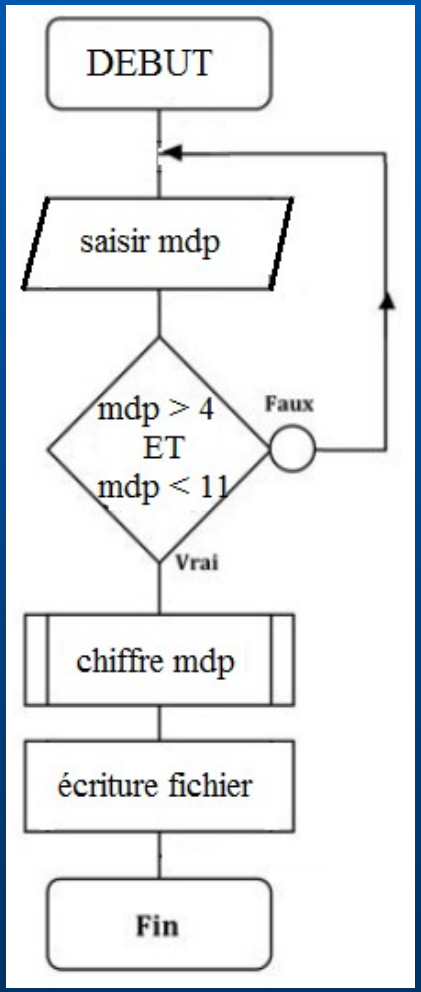
* avec un algorithme de chiffrement par décalage

Introduction à l'algorithmique

Informatique et Science du Numérique

Notions abordées

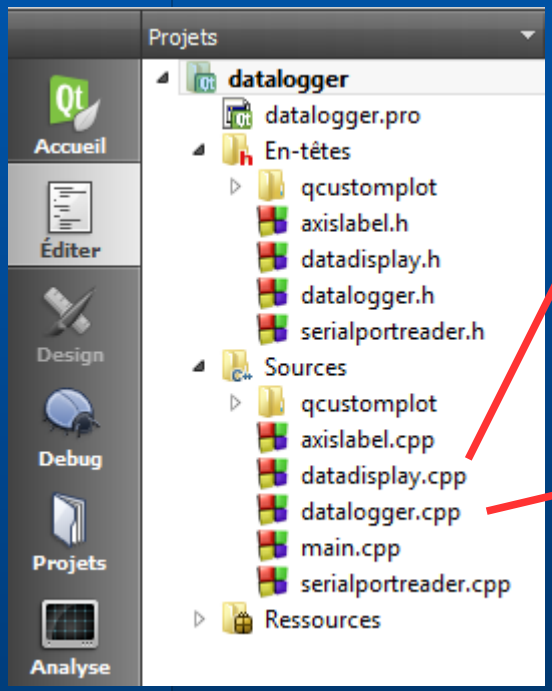
- Programmation modulaire
- Fonctions statiques
- Fichier à accès aléatoire



Introduction à l'algorithmique

Informatique et Science du Numérique

Programmation modulaire




```
datadisplay.cpp

void setup()
{
   //...
}
```

Collision !

```
datalogger.cpp

void setup()
{
   //...
}
```

Utiliser le mot clef stastic

Introduction à l'algorithmique

Informatique et Science du Numérique

Exemple 4 : le tri à bulles

Mesurer la performance de l'algorithme du tri à bulles* sur une liste de 1000 éléments.

- Créer une liste de 1000 entiers [0 ; 500] générés aléatoirement.
- Trier la liste selon l'algorithme du tri à bulles.
- Afficher le temps de calcul.
- Enregistrer la liste triée dans un fichier.

Bonus :

- Enregistrer le temps de traitement pour des listes de tailles différentes.
- Évaluer graphiquement la performance de l'algorithme.

* permutations successives d'éléments adjacents

Introduction à l'algorithmique

Informatique et Science du Numérique

Exemple 5 : occurrence des lettres

Déterminer l'occurrence des lettres de l'alphabet dans un texte*.

- Utiliser un fichier comportant au moins 1000 caractères.
- Afficher les occurrences en %.
- Afficher les occurrences avec un histogramme.

Bonus :

- Afficher les occurrences en % par ordre décroissant.

* utilisé en cryptographie pour le déchiffrement par méthode statistique

Introduction à l'algorithmique

Informatique et Science du Numérique

Exemple 6 : pour terminer...

Écrire un programme pour rechercher les racines d'un polynôme :

- Quelconque de degré 2
- D'équation : $y = 0{,}25.x^3 + 0{,}75.x^2 - 1{,}5.x - 1{,}9$