

Notions de base

Cours Algorithmique Sup

mercredi 7 septembre 2011

Quelques définitions

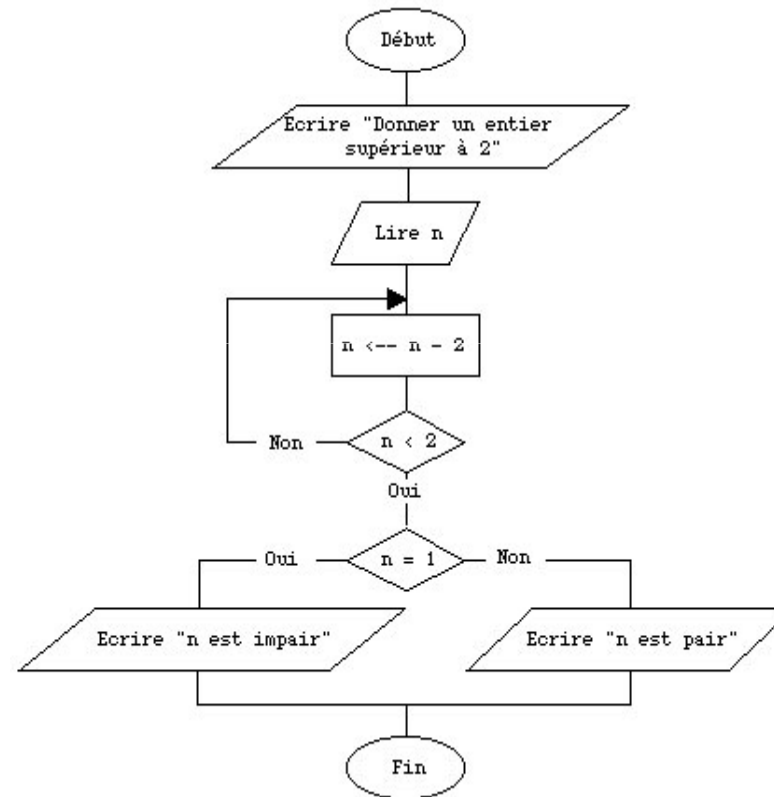
- Informatique : **information & automatique**
- Ordinateur :
- Algorithme :
- Langage :
- Compilateur :
- Procédural :

Algorithme

- suite ordonnée d'opérations produisant un résultat en un temps fini

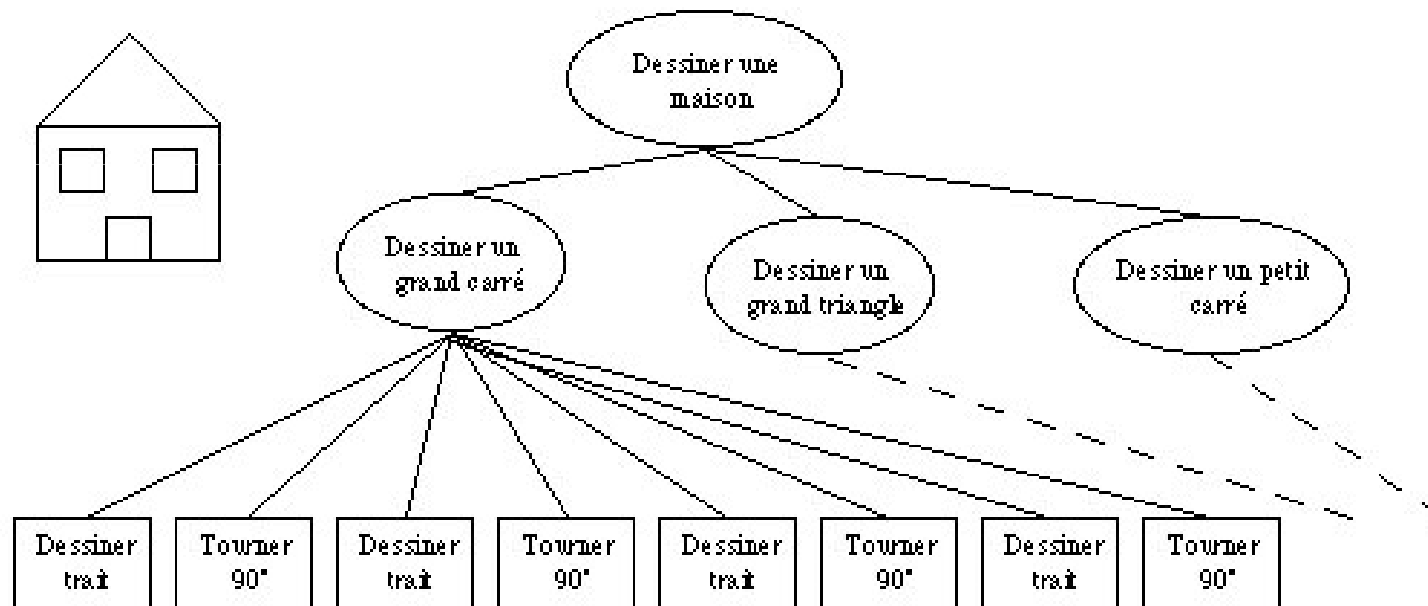
Formalisme

- Organigramme



Formalisme

- Arbre programmatique :



Formalisme

■ Pseudo-code :

Début

Lire x et y

Initialiser P à 0

Tant que y > 0 Faire

 P ← P + x

 Y ← y - 1

Fin Tant que

Ecrire P

Fin

Quelques consignes

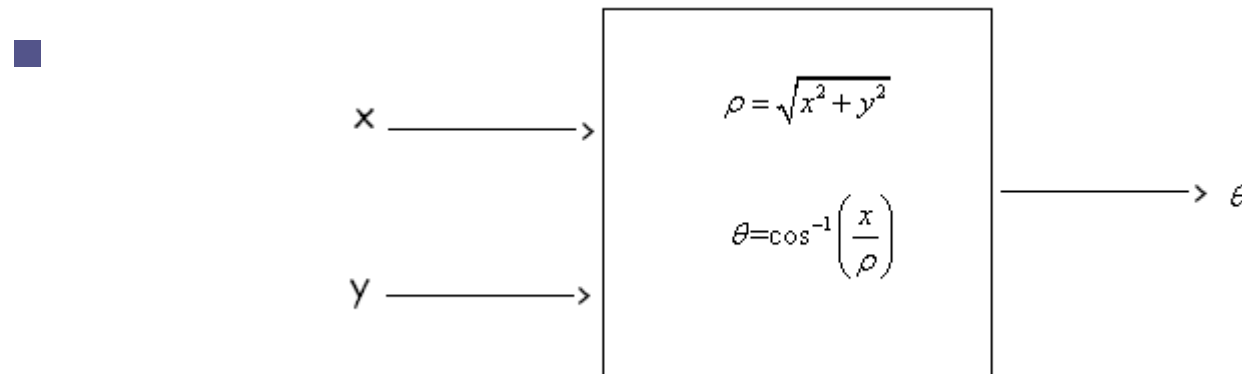
- faire des arbres programmatiques dès que notre algorithme est complexe ;
- bien indenter notre programme ;
- donner des noms mnémoniques à nos composants (variable, procédure,...) ;
- commenter nos programmes (surtout dans le cas d'utilisation d'astuces de programmation).

Informations manipulées , Données

- Les constantes :
- Les variables et types de données simples : entiers, réels, booléens, caractères, chaînes
- Les tableaux :
- Les enregistrements, fichiers, bases de données :

Méthode

- Pensez à faire un bilan des données manipulées avant de commencer à produire un algorithme



Structures algorithmiques

Structures conditionnelles

Structures itératives :

nombre d'itérations inconnu

répéter ...jusqu'à

tant que ...faire

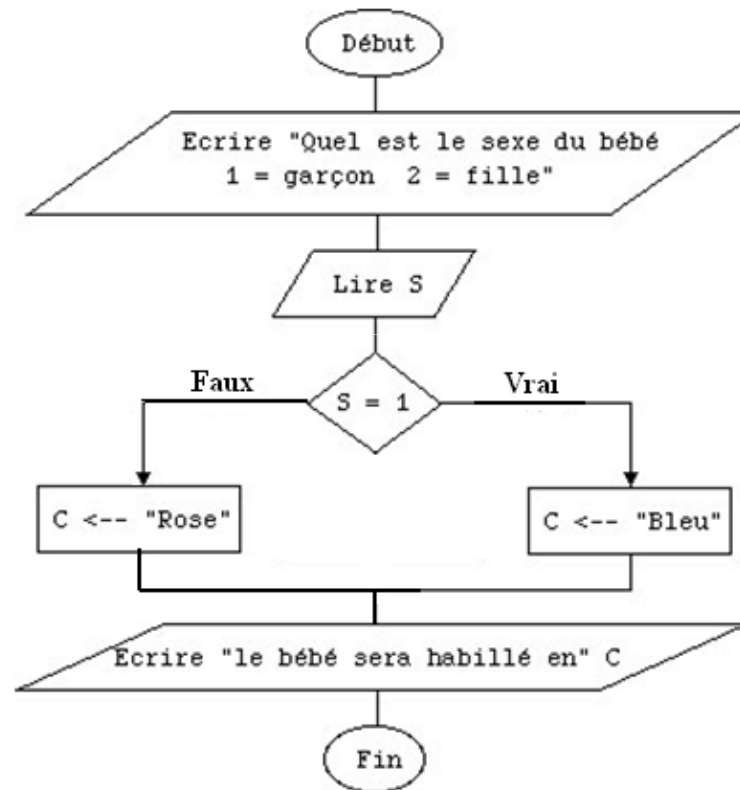
nombre d'itérations connu

pour...faire

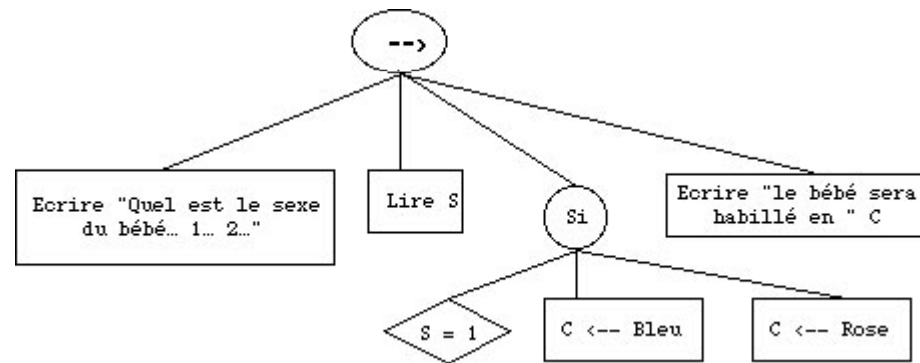
Procédures et fonctions

- Gérer l'architecture de programmes plus complexes, plus longs
- Réutiliser des développements précédents
- Partitionner le travail

Structure conditionnelle : organigramme



Structure conditionnelle : arbre programmatique



mercredi 7 septembre 2011

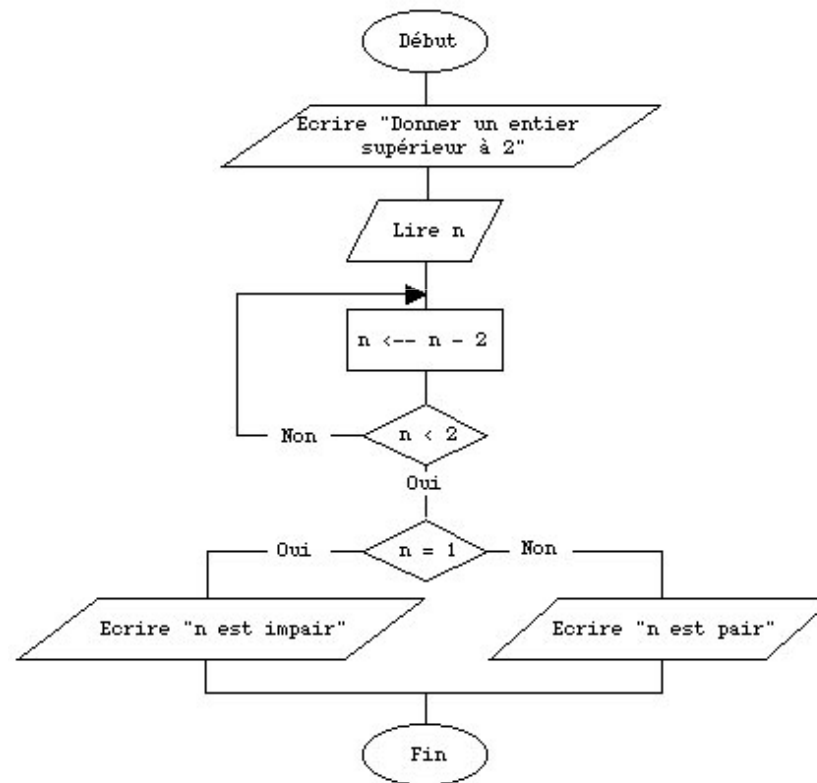
Structure conditionnelle : pseudo-code

```
Si <test>  
  alors <instr.1>  
  sinon <instr.2>  
FinSi
```

En Pascal:

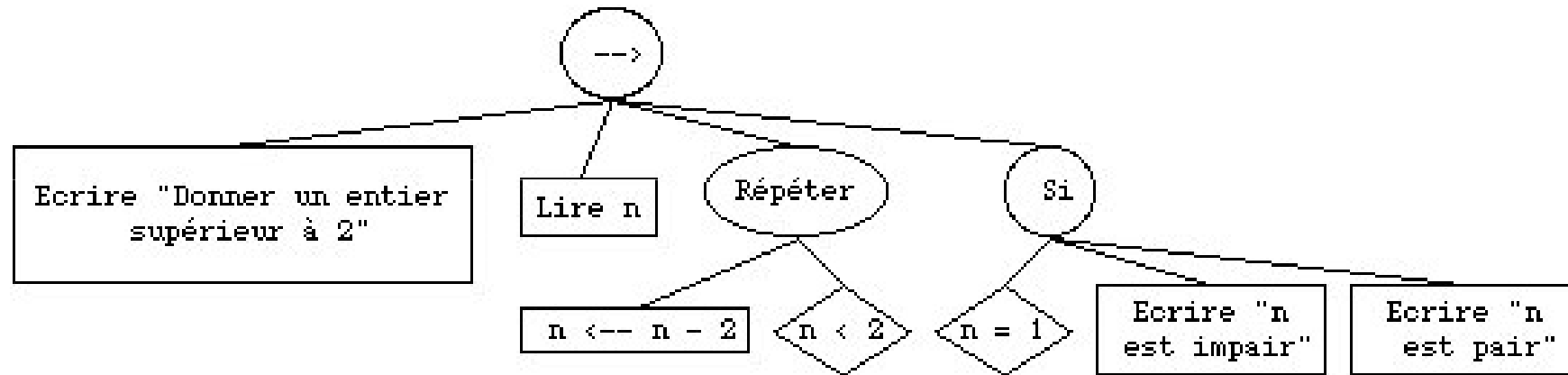
```
If <test>  
  Then  
    Begin  
      <instr.1>;  
      <instr.2>;  
      ...;  
    End  
  Else  
    Begin  
      <instr.a>;  
      <instr.b>;  
      ...;  
    End;
```

Structure Répéter ... jusqu'à



mercredi 7 septembre 2011

Structure Répéter ... jusqu'à



Structure Répéter ... jusqu'à

Début

Ecrire "Donner un entier supérieur à 2"

Lire n

Répéter

$n \leftarrow n - 2$

Jusqu'à $n < 2$

Si $n = 1$

Alors Ecrire "n est impair"

Sinon Ecrire "n est pair"

FinSi

Fin.

Structure Répéter ... jusqu'à

Repeat

<instr. 1>;

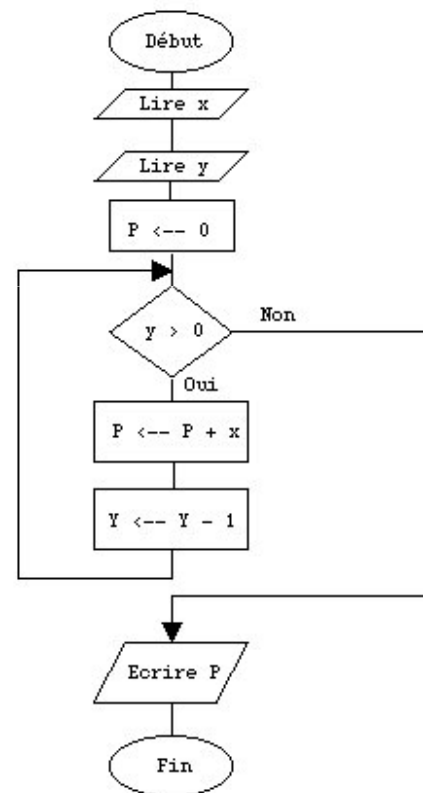
<instr. 2>;

...

<instr. n>;

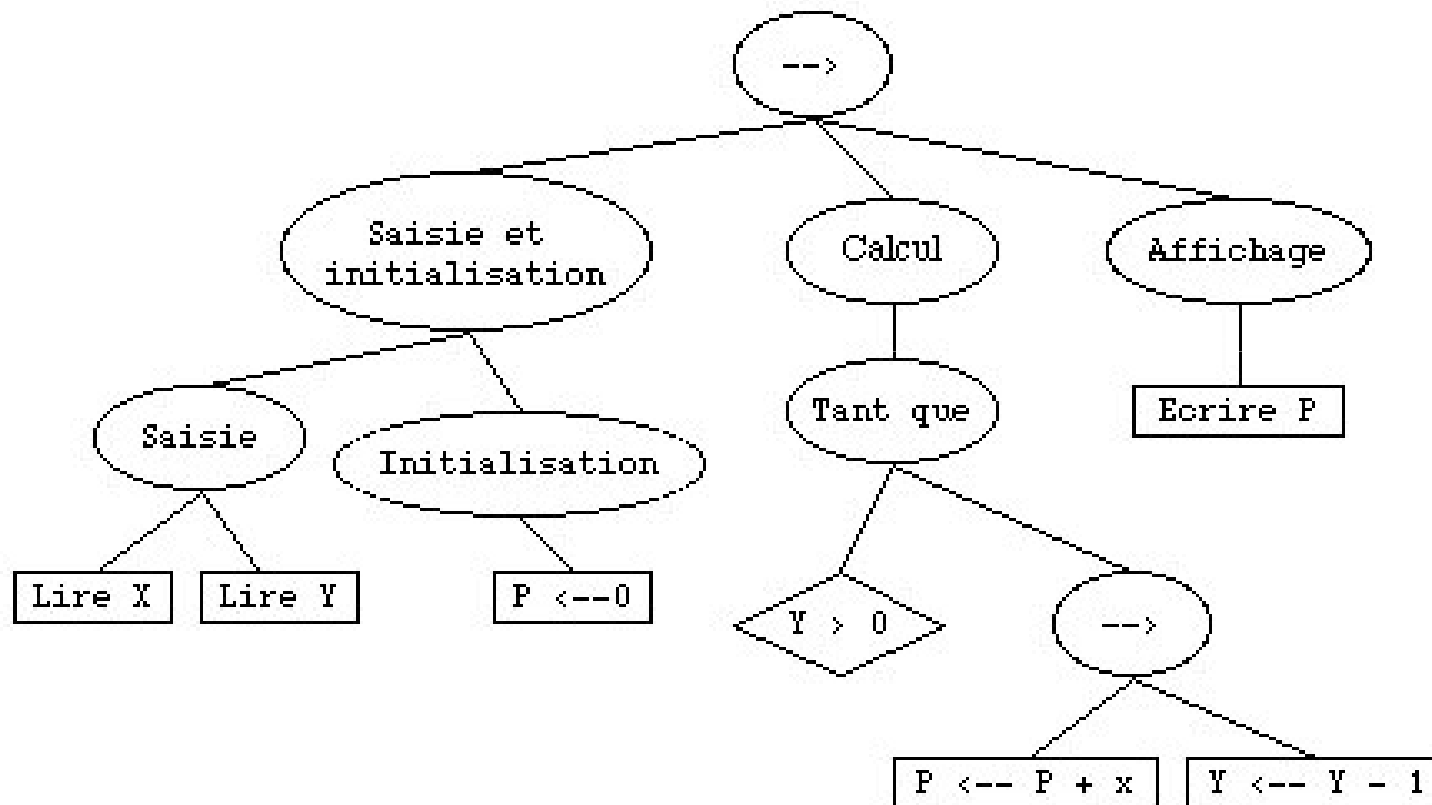
Until <test>;

Structure Tant que ... faire



mercredi 7 septembre 2011

Structure Tant que ... faire



Structure Tant que ... faire

Début

Lire x et y

Initialiser P à 0

Tant que y > 0 Faire

P \leftarrow P + x

Y \leftarrow y - 1

Fin Tant que

Ecrire P

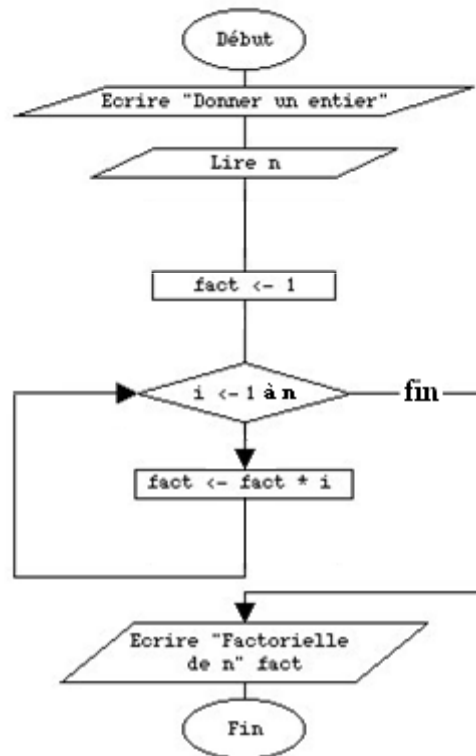
Fin

Structure Tant que ... faire

```
While <test> do  
  <instr.1>;
```

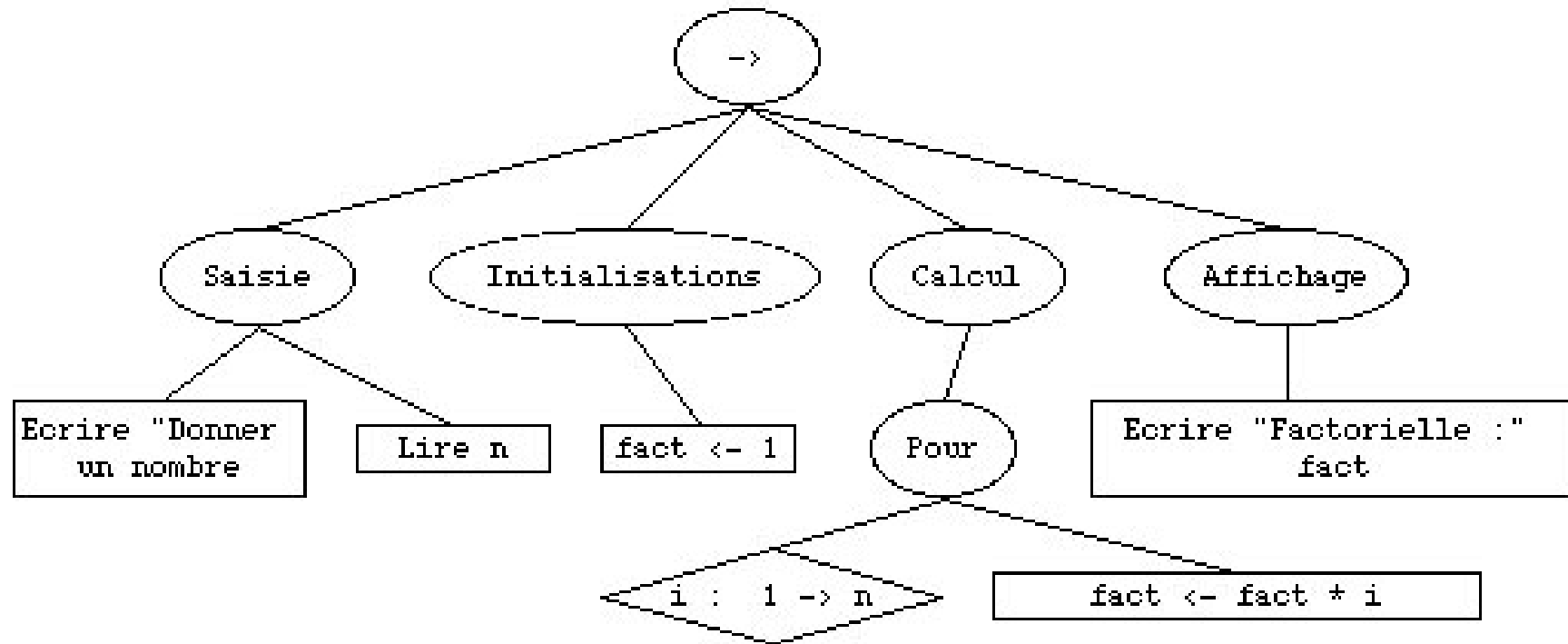
```
While <test> do  
  begin  
    <instr.1>;  
    <instr.2>;  
    ...;  
  end;
```

Structure Pour ... faire



mercredi 7 septembre 2011

Structure Pour ... faire



Structure Pour ... faire

Début

Ecrire ("Donner un entier")

Lire n

fact <- 1

Pour tout i allant de 1 à n faire

fact <- fact * i

Fait (ou FinFaire)

Ecrire ("Factorielle de n :", fact)

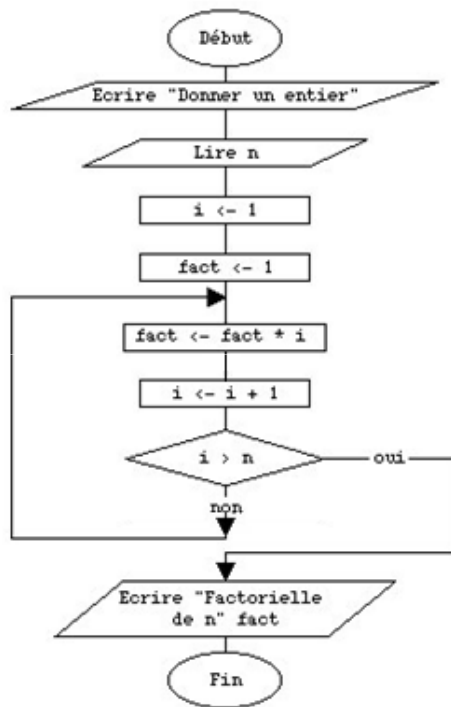
Fin.

Structure Pour ... faire

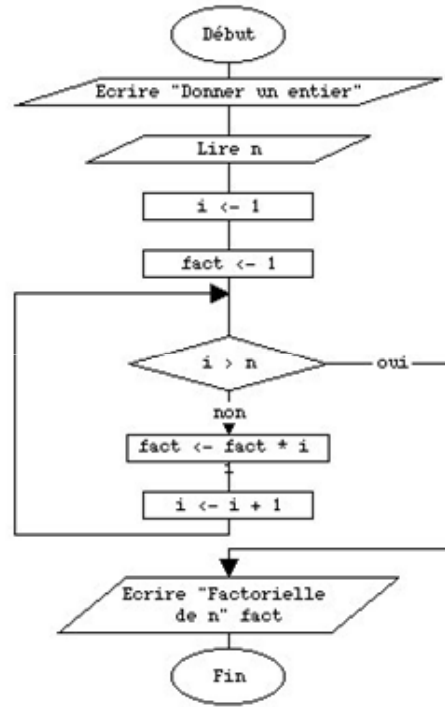
```
For <compteur> := <min> to  
  <max> do  
  <instr.>;
```

```
For <compteur> := <min> to  
  <max> do  
  begin  
    <instr.1>;  
    <instr.2>;  
    ...  
  end;
```

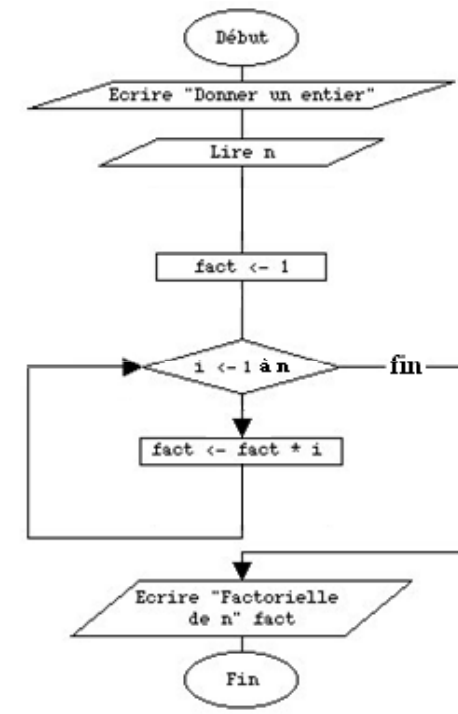
Comparaison entre les 3 boucles



Répéter ... jusqu'à



Tant que ... faire



Pour ... faire