

Initiation à l'Algorithmique

Algorithmique

Structures répétitives

- ▶ Ex : afficher des nombres de 1 à 1000

Algo affiche
Début
 écrire(1); écrire(2);
 ; ;
 écrire(999); écrire(1000);
Fin.
Notre code contient 1000
instructions
Solution : Utiliser les
boucles



Algo affiche
Var
 i : entier;
Début
 i ← 1;
 Tantque (i ≤ 1000) faire
 écrire (i);
 i ← i + 1;
 FinTq;
Fin.

Les boucles permettent d'exécuter un bloc d'instructions
plusieurs fois

Le nombre de fois peut être connu ou dépend d'une condition.

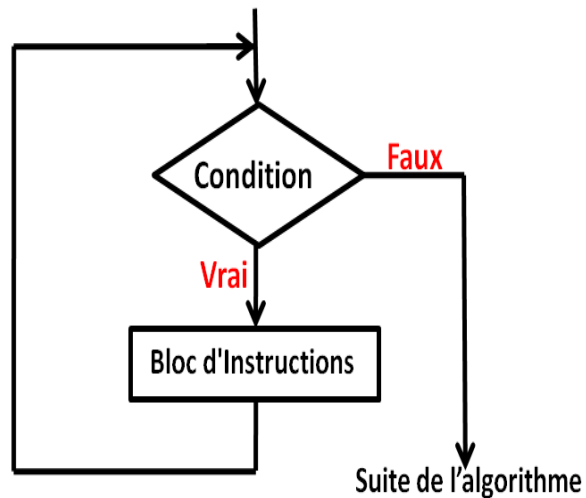
Structures répétitives

- ▶ Trois types de boucles :

Algorithmique	en C++
Tantque (Condition) faire Bloc d'instructions; FTq	while (expression) { Bloc d'instructions; }
Faire (Répéter) Bloc d'instructions; tant que (condition);	do { Bloc d'instructions; } while (condition);
Pour (idvar de VI à VF pas=p) faire Bloc d'instructions; FinPour	For (idvar=VI; idvar<VF; idvar+=p) { Bloc d'instructions; }

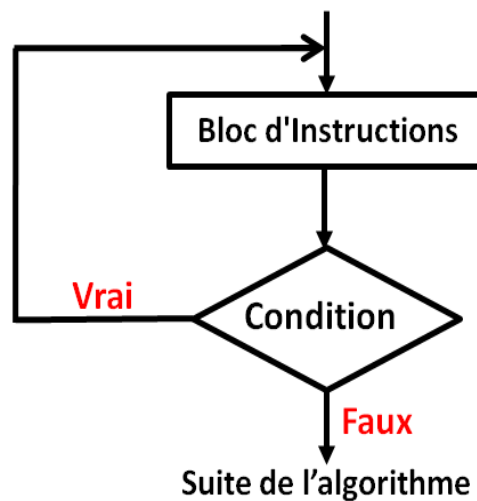
Organigramme

Tant que



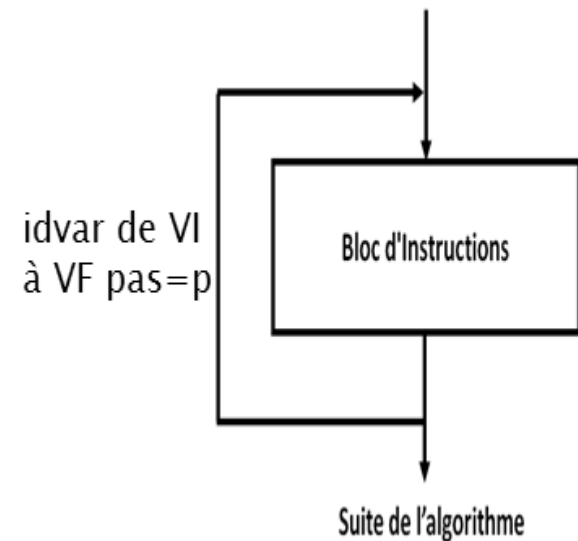
Vérifier la condition
avant d'exécuter le bloc

Faire (Répéter)



Exécuter le bloc puis
vérifier la condition

Pour (idvar de VI
à VF pas=p)



Exécuter le bloc
plusieurs fois selon VI
et VF et p

Exemples

- ▶ Exemple 1 : écrire un algorithme qui demande à l'utilisateur de saisir un nombre N puis affiche les nombres de 0 à N.
- ▶ Puisqu'on connaît le nombre d'itérations : utiliser la boucle Pour

Algo affiche

Var

I, N : entier;

Début

Écrire ("Donner un nombre:");

Lire(N);

Pour(I de 0 à N) faire

Écrire (I, "\n");

Finpour;

Fin.

NB : si on ne mentionne pas le pas
ce dernier est par défaut = 1

En C++ :

Int main()

{

int I, N;

cout << "Donner un nombre:";

cin >> N;

for(I=0; I<=N; I++)

cout << I << "\n";

}

Exemples

- ▶ Exemple 2 : écrire un algorithme qui demande à l'utilisateur de saisir un nombre N puis affiche tous les nombres pairs inférieurs à N.
ex N=8 donc afficher : 0, 2, 4, 6,

Algo affiche2

Var

I, N : entier;

Début

Écrire ("Donner un nombre:");

Lire(N);

Pour(I de 0 à N-1 pas=2) faire

Écrire (I, ", ");

Finpour;

Fin.

En C++ :

Int main()

{

int I, N;

cout << "Donner un nombre:";

cin >> N;

for(I=0; I<N; I+=2)

cout << I << ", ";

}

Exemples

- ▶ Exemple 3 : Ecrire un algorithme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne. En cas où la réponse est bonne on affiche le message : Bonne Réponse.
- ▶ Puis dresser l'organigramme
- ▶ N compris entre 1 et 3 : $1 \leq N \leq 3$: en Algo ($N \leq 3$) et ($N \geq 1$))

Algo choix

Var

N : entier;

Début

faire

écrire ("Donner un nombre
entre 1 et 3:");

lire (N);

Tant que ((N<1) ou (N>3));

écrire ("Bonne Réponse:");

Fin.

En C++ :

Int main()

{

int N;

do

{

cout << "Donner un nombre";

cin >> N;

}

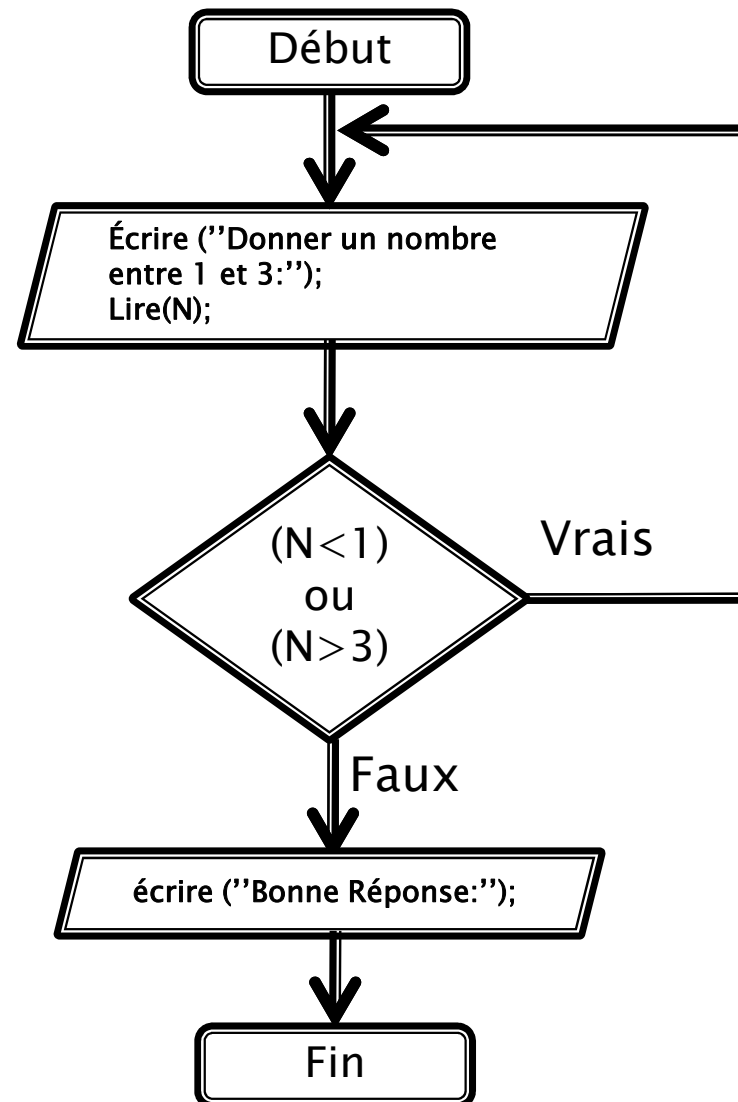
while ((N<1) || (N>3));

cout << "Bonne Réponse";

}

Exemples

- ▶ Exemple 2 : l'organigramme



Exemples

- ▶ Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.

Algo choix

Var

N : entier;

Début

Faire

Ecrire ("Entrez un numéro:");

Lire (N);

Si (N < 10) Alors

Ecrire ("Plus grand !");

Sinon

Si (N > 20) Alors

Ecrire ("Plus petit !");

FinSi;

FinSi;

Tant que ((N > 20) ou (N < 10));

Écrire ("Vous avez tapé un bon numéro!");

Fin.

En C++ :

```
int main() {
```

```
    int N;
```

```
    do
```

```
    {
```

```
        cout << "Donner un nombre:";
```

```
        cin >> N;
```

```
        if (N < 10)
```

```
            cout << "plus grand!\n";
```

```
        else
```

```
            if (N > 20)
```

```
                cout << "plus petit!\n";
```

```
        }
```

```
        while ((N<10) || (N>20));
```

```
        cout << "Vous avez tape un bon\n\n";
```

```
    }
```

Exemples

- ▶ Ecrire un algorithme qui demande à l'utilisateur d'entrer un nombre puis calcule et affiche le factoriel de ce nombre (en utilisant tant que)

Algo Fact

Var

F, N : entier;

Début

Ecrire ("Entrez un nombre:");

Lire (N);

F ← 1;

Tant que (N > 1)

F ← F * N;

N ← N - 1;

Fin TQ;

Ecrire ("le Factorielle =", F);

Fin.

En C++ :

```
int main() {  
    int N,F;  
    F=1;  
    cout << "Entrez un nombre : ";  
    cin >> N;  
    while (N>1)  
    {  
        F=F*N,  
        N--;  
    }  
    cout << "le FACT="<<F ;  
}
```

Exemples

- ▶ Autre solution en utilisant la boucle pour :

Algo Fact

Var

F, N, i : entier;

Début

Ecrire ("Entrez un nombre:");

Lire (N);

$F \leftarrow 1$;

pour (i de 2 à N) faire

$F \leftarrow F * i$;

Fin pour;

Ecrire ("le Factorielle =", F);

Fin.

En C++ :

```
int main()
```

```
{
```

```
    int i,N,F;
```

```
    cout << "Entrer un nombre : ";
```

```
    cin >> N;
```

```
    F=1;
```

```
    for (i=2; i<=N; i++)
```

```
        F=F*i,
```

```
    cout << "le FACT="<<F ;
```

```
}
```