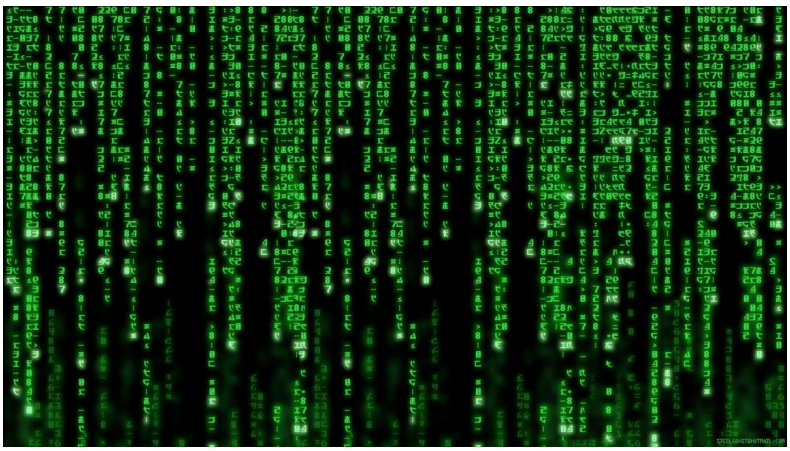
UVSQ université PARIS-SACLAY | IUT de Vélizy-Rambouillet CAMPUS DE VÉLIZY-VILLACOUBLAY

Chapitre 4 : Structures Répétitives



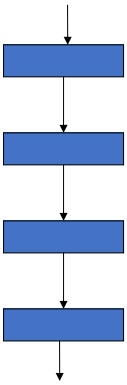
GEII 1ère année
IUT Vélizy
2022 – 2023
Julien Gabiot

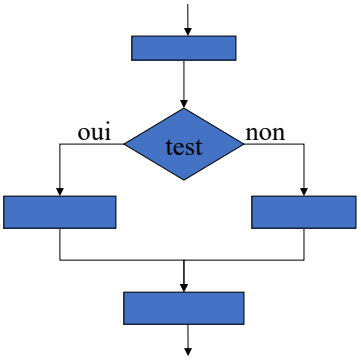
1

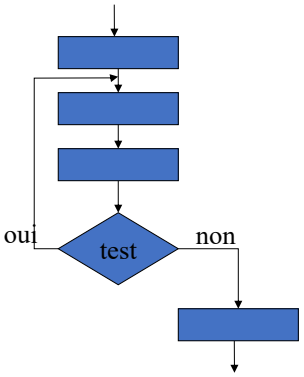
Introduction

UVSQ université PARIS-SACLAY | IUT de Vélizy-Rambouillet CAMPUS DE VÉLIZY-VILLACOUBLAY

- Algorithme séquentiel : instructions toutes effectuées à la suite
- Algorithme conditionnel : instructions effectuées sous certaines conditions
- Algorithme répétitif : instructions répétées un certain nombre de fois







2

Introduction

UVSQ université PARIS-SACLAY | IUT de Vélizy-Rambouillet CAMPUS DE VÉLIZY-VILLACOUBLAY

Classification des structures répétitives (ou itératives ou boucles)

Boucles prédéfinies (ou inconditionnelles) : nombre de tours de boucle connu

➔structure **for**

Boucles conditionnelles (ou indéfinies) : instructions répétées tant qu'une condition est vérifiée

➔ structures **while** et **do … while**

Test à priori : la condition est testée avant d'exécuter les instructions

➔ structures **for** et **while**

Test à posteriori : la condition est testée à la fin

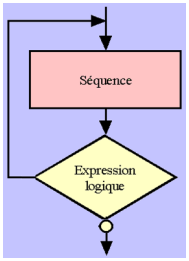
➔ structure **do … while**

3

Boucles Conditionnelles

UVSQ université PARIS-SACLAY | IUT de Vélizy-Rambouillet CAMPUS DE VÉLIZY-VILLACOUBLAY

La structure Faire… Tant que (`do… while`)



- instruction : instruction simple, bloc d'instructions ou instruction conditionnelle
- instructions effectuées avant le test : au moins *un tour de boucle*
- condition jamais fausse, on ne sort pas de la boucle ➔ programme infini

- En C: **do… while**

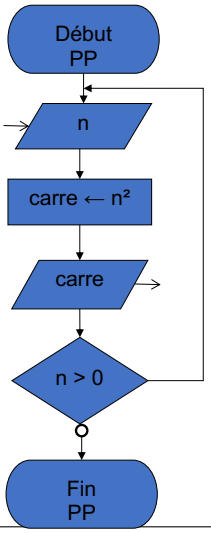
```
do
   instruction;
while (condition);

do {
    instruction1;
    instruction2;
    :
    :
    instructionN;
} while (condition);
```


4

Boucles conditionnelles

UVSQ université PARIS-SACLAY | IUT de Vélizy-Rambouillet CAMPUS DE VÉLIZY-VILLACOUBLAY

Exemple : calcul du carré d'un entier tant qu'il est positif




```
#include <stdio.h>
int main(void) {
  int n, carre;
  do {
    printf("entrez un nombre entier : ");
    scanf("%d",&n);
    carre = n * n;
    printf("son carre vaut %d\n",carre);
  } while (n>0);
}
```

Exemple d'exécution :

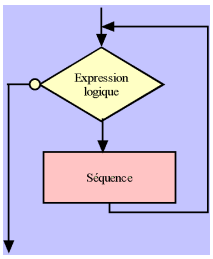
```
entrez un nombre entier : 6
son carré vaut 36
entrez un nombre entier : 4
son carré vaut 16
entrez un nombre entier : -5
son carré vaut 25
```

5

Boucles conditionnelles

UVSQ université PARIS-SACLAY | IUT de Vélizy-Rambouillet CAMPUS DE VÉLIZY-VILLACOUBLAY

La structure Tant que (`while`)



- instruction : instruction simple, bloc d'instructions ou instruction conditionnelle
- instructions effectuées après le test : on peut ne *jamais* rentrer dans la boucle
- condition jamais fausse, on ne sort pas de la boucle ➔ programme infini

- En C: **while**

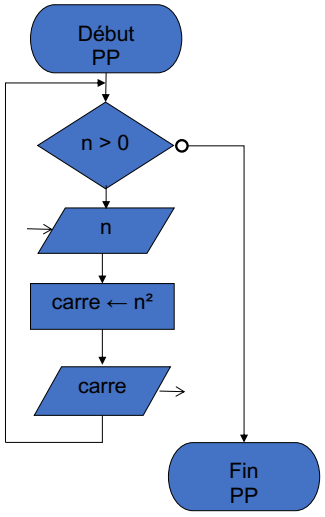
```
while (condition)
  instruction;

while (condition) {
   instruction1;
   instruction2;
   :
   :
   instructionN;
}
```

6

2. Boucles conditionnelles

UVSQ université PARIS-SACLAY — IUT de Vélizy-Rambouillet, CAMPUS DE VÉLIZY-VILLACOUBLAY

Exemple : calcul du carré d'un entier tant qu'il est positif




```
#include <stdio.h>
int main(void) {
  int n, carre;
  n = 1 ;
  while (n>0) {
    printf("entrez un nombre entier : ");
    scanf("%d",&n);
    carre=n * n;
    printf("son carre vaut %d\n",carre);
  }
  return 0 ;
}
```

n *non initialisée*

➔ initialisation arbitraire
(ex : n = 1)

7

Boucles Prédéfinies

UVSQ université PARIS-SACLAY — IUT de Vélizy-Rambouillet, CAMPUS DE VÉLIZY-VILLACOUBLAY

La structure Pour… Fin Pour (for)

- Nombre de tours connu ➔ variable entière n : nombre total de tours à effectuer
- Nécessité de compter les tours ➔variable entière compteur (ou i, j …)
  - valeur initiale : 0 (ou autre) / valeur finale : 10 (ou autre)
  - Incrémentation du compteur : 1 (le pas)

Initialisation — Condition / valeur finale — Pas / incrémentation

```
for (i=0;i<=10;i=i+1)
   instruction;
```

```
for (i=0;i<=10;i=i+1)
{
   instruction1;
   instruction2;
   :
   :
   instructionN;
}
```

Ne pas modifier la valeur du compteur (*i.e.* i)
à l'intérieur de la boucle

8

Boucles Prédéfinies

UVSQ université PARIS-SACLAY — IUT de Vélizy-Rambouillet, CAMPUS DE VÉLIZY-VILLACOUBLAY

- ✓ En général : ideb = 0 (ou 1), ifin = n-1 (ou n), pas : +1

```
for (i=ideb;i<=ifin;i=i+pas)
   instruction;
```

- ✓ Instruction d'incrémentation : `i++` ≡ `i = i+1`

~~`i = i++ ;` ≡ `i = i = i+1 ;`~~

- ✓ Boucle `for` standard :

```
for (i=0;i<n;i++)
   instruction;
```

ou

```
for (i=1;i<=n;i++)
   instruction;
```

- ✓ Instruction de décrémentation : `i--` ≡ `i = i-1`

```
for (i=n-1;i>=0;i--)
   instruction;
```

```
for (i=n;i>0;i--)
   instruction;
```

Boucles prédéfinies

UVSQ université PARIS-SACLAY — IUT de Vélizy-Rambouillet, CAMPUS DE VÉLIZY-VILLACOUBLAY

Exemple : afficher les tours de boucles

```
#include <stdio.h>

int main(void) {
  int i,n;
  printf("nombre de tours : ");
  scanf("%d",&n);
  for (i=1;i<=n;i++) {
    printf("tour n°%d\n",i);
  }
}
```

Exemples d'exécution :

```
entrez un nombre entier : 6
tour n°1        (i=0)
tour n°2        (i=1)
tour n°3        (i=2)
tour n°4        (i=3)
tour n°5        (i=4)
tour n°6        (i=5)
                (i=6)

      -----------------

entrez un nombre entier : 0
                (i=0)
```

Boucles prédéfinies

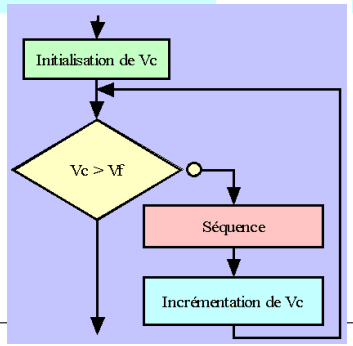
UVSQ université PARIS-SACLAY — IUT de Vélizy-Rambouillet, CAMPUS DE VÉLIZY-VILLACOUBLAY

Remarque : En C une boucle `for` *est strictement équivalente à une boucle* `while`

```
for (initialisation;condition;
   incrémentation)
  instruction;
```

=

```
initialisation;
while (condition) {
  instruction;
  incrémentation;
}
```



11

Instructions de contrôle

UVSQ université PARIS-SACLAY — IUT de Vélizy-Rambouillet, CAMPUS DE VÉLIZY-VILLACOUBLAY

- Permet de gérer les exceptions
  - ➔ sortir directement de la boucle
  - ➔ recommencer directement un nouveau tour de boucle
- Placées à l'intérieur d'une structure conditionnelle (`if`)
- Sortir de la boucle

break;

- Recommencer un nouveau tour

continue;

```
#include <stdio.h>
int main(void) {
  int n, carre;
  while (1) {
    printf("entrez un nombre entier : ");
    scanf("%d",&n);
    if (n<0)
        break;
    carre=n * n;
    printf("son carre vaut %d\n",carre);
  }
  return 0;
}
```

12

Algorithmes Classiques

UVSQ université PARIS-SACLAY | IUT de Vélizy-Rambouillet CAMPUS DE VÉLIZY-VILLACOUBLAY

Somme

- Initialiser une variable `somme` à 0 avant la boucle
- À chaque tour de boucle, incrémenter `somme` de la valeur voulue

Exemple : somme de 10 valeurs rentrées par l'utilisateur

```
#include <stdio.h>

int main(void) {
  int somme,i,val;
  printf("entrez les 10 valeurs");
  somme = 0;
  for (i=0;i<10;i++) {
    scanf("%d",&val);
    somme = somme + val;
    //ou somme += val;
  }
  printf("la somme vaut %d\n", somme);
}
```

Boucles imbriquées

UVSQ université PARIS-SACLAY | IUT de Vélizy-Rambouillet CAMPUS DE VÉLIZY-VILLACOUBLAY

- Plusieurs boucles ou structures conditionnelles peuvent être successives ou imbriquées mais pas croisées
- À chaque tour de la boucle « extérieure », on effectue plusieurs tours de boucle « intérieure »
- Si 2 boucles prédéfinies (`for`) sont imbriquées, il faut utiliser 2 compteurs différents

```
for (i=0;i<10;i++) {
  printf("Bidule");
  for (i=0;i<n;i++) {
    printf("Truc");
  }
}
```

```
n = 20 ➔ 1 Bidule, 20 Truc
n = 5  ➔ boucle infinie
```

```
for (i=0;i<10;i++) {
  printf("Bidule");
  for (j=0;j<n;i++) {
    printf("Truc");
  }
}
```

```
➔ 10 Bidule
➔ 66 Truc
```