

# IN100 : Cours 4 – Les Boucles

Sandrine Vial

`sandrine.vial@prism.uvsq.fr`

Octobre 2015

# Rappel

## La semaine prochaine

- ▶ **Mercredi 14 Octobre de 09h45 à 11h15** : **Contrôle Continu** en amphi D et en amphi F.
  - ▶ QCM + 1 ou 2 questions ouvertes
  - ▶ **Seul document autorisé** : une feuille A4 recto-verso avec ce que vous voulez dessus
  - ▶ **Aucun** appareil électronique autorisé.

## La semaine du 19 Octobre

- ▶ **Durant votre séance habituelle de TD** : **Khôlle**.
  - ▶ Amener un ordinateur (le vôtre ou un cartable numérique).
  - ▶ Documents autorisés : vos cours et vos codes écrits en TD.

# Les boucles

- ▶ Toutes les lignes de code sont exécutées séquentiellement.
- ▶ **Embranchement** (`if (...) ... else ....`) : Permet de choisir les lignes de codes à exécuter en fonction d'une condition.
- ▶ **Boucles** : Permet de **répéter** un *bloc d'instructions* en fonction d'*une condition*.

# L'instruction while

Tant qu'une *condition* est remplie, exécuter un *bloc d'instructions*.

```
while (condition)  
{  
    bloc d'instructions  
}
```

# Déroulement des instructions

```
int i;
write_text("Entrez un entier > 0");
i = lire_entier_clavier();

while (i <= 0)
{
    write_text("Erreur!");
    write_text("Entrez un entier > 0");
    i = lire_entier_clavier();
}
write_text("Merci!");
```

Suite d'Instructions Exécutées

# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");

# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");
2. i = lire\_entier\_clavier();

i = -12

# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");
2. i = lire\_entier\_clavier();
3. while (i <= 0)

i = -12



# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");
2. i = lire\_entier\_clavier();
3. while (i <= 0)
4. write\_text("Erreur!");  
 write\_text("Entrez un entier > 0");  
 i = lire\_entier\_clavier();

i = 56

# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

**i = 56**

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");
2. i = lire\_entier\_clavier();
3. while (i <= 0)
4. write\_text("Erreur!");  
write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();
5. while (i <= 0)

# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

**i = 56**

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");
2. i = lire\_entier\_clavier();
3. while (i <= 0)
4. write\_text("Erreur!");  
write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();
5. while (i <= 0)
6. write\_text("Merci!");

# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. `write_text("Entrez un entier > 0");`
2. `i = lire_entier_clavier();`
3. `while (i <= 0)`
4. `write_text("Erreur!");`  
`write_text("Entrez un entier > 0");`  
`i = lire_entier_clavier();`
5. `while (i <= 0)`
6. `write_text("Merci!");`

**i = 56**

# Déroulement des instructions

```
int i;
write_text("Entrez un entier > 0");
i = lire_entier_clavier();

while (i <= 0)
{
    write_text("Erreur!");
    write_text("Entrez un entier > 0");
    i = lire_entier_clavier();
}
write_text("Merci!");
```

Suite d'Instructions Exécutées

# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");

# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");
2. i = lire\_entier\_clavier();

i = 658

# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");
2. i = lire\_entier\_clavier();
3. while (i <= 0)

i = 658



# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");
2. i = lire\_entier\_clavier();
3. while (i <= 0)
4. write\_text("Merci!");

i = 658

# Déroulement des instructions

```
int i;  
write_text("Entrez un entier > 0");  
i = lire_entier_clavier();  
  
while (i <= 0)  
{  
    write_text("Erreur!");  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
}  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");
2. i = lire\_entier\_clavier();
3. while (i <= 0)
4. write\_text("Merci!");

i = 658

## Example

```
POINT p1,p2;

init_graphics(600,300);

p1.x = 10;
p2.x = p1.x;
p1.y = 10;
p2.y = 290;
while (p1.x < 590)
{
    draw_line(p1,p2,bleu);
    p1.x = p1.x + 40;
    p2.x = p1.x;
}
wait_escape();
exit(0);
```

# Attention Pièges!

## Boucle infinie

```
int i;  
  
i = 12;  
while (i > 0)  
{  
    write_int(i);  
    i = i + 1;  
}
```

## Boucle sans effet

```
int i;  
  
i = 12;  
while (i < 0)  
{  
    write_int(i);  
    i = i + 1;  
}
```

# Attention Pièges!

```
int i;  
  
i = lire_entier_clavier();  
while (i < 0);  
{  
    write_text("Entrez un entier >0);  
    i = lire_entier_clavier();  
}
```

# L'Instruction for

## Deux définitions

1. Après *initialisation* et tant qu'une *condition* est remplie, exécuter un *bloc d'instructions*.
2. Pour une valeur comprise entre une *valeur minimale* et une *valeur maximale* faire *bloc d'instructions*.

```
for (initialisation; condition; poursuite)  
{  
    bloc d'instructions  
}
```

# Déroulement des instructions

```
int i, somme;  
  
somme = 0;  
for (i = 1; i < 3; i++)  
{  
    somme = somme + i;  
}  
write_int(i);
```

Suite d'Instructions Exécutées

# Déroulement des instructions

```
int i, somme;
```

```
somme = 0;
```

```
for (i = 1; i < 3; i++)
```

```
{
```

```
    somme = somme + i;
```

```
}
```

```
write_int(i);
```

## Suite d'Instructions Exécutées

1. somme = 0;



# Déroulement des instructions

```
int i, somme;  
  
somme = 0;  
for (i = 1; i < 3; i++)  
{  
    somme = somme + i;  
}  
write_int(i);
```

## Suite d'Instructions Exécutées

1. somme = 0;
2. i = 1;

# Déroulement des instructions

```
int i, somme;  
  
somme = 0;  
for (i = 1; i < 3; i++)  
{  
    somme = somme + i;  
}  
write_int(i);
```

## Suite d'Instructions Exécutées

1. somme = 0;
2. i = 1;
3. i < 3;

# Déroulement des instructions

```
int i, somme;  
  
somme = 0;  
for (i = 1; i < 3; i++)  
{  
    somme = somme + i;  
}  
write_int(i);
```

## Suite d'Instructions Exécutées

1. `somme = 0;`
2. `i = 1;`
3. `i < 3;`
4. `somme = somme + i;`

# Déroulement des instructions

```
int i, somme;  
  
somme = 0;  
for (i = 1; i < 3; i++)  
{  
    somme = somme + i;  
}  
write_int(i);
```

## Suite d'Instructions Exécutées

1. `somme = 0;`
2. `i = 1;`
3. `i < 3;`
4. `somme = somme + i;`
5. `i++`

# Déroulement des instructions

```
int i, somme;  
  
somme = 0;  
for (i = 1; i < 3; i++)  
{  
    somme = somme + i;  
}  
write_int(i);
```

## Suite d'Instructions Exécutées

1. somme = 0;
2. i = 1;
3. i < 3;
4. somme = somme + i;
5. i++
6. i < 3;

# Déroulement des instructions

```
int i, somme;  
  
somme = 0;  
for (i = 1; i < 3; i++)  
{  
    somme = somme + i;  
}  
write_int(i);
```

## Suite d'Instructions Exécutées

1. `somme = 0;`
2. `i = 1;`
3. `i < 3;`
4. `somme = somme + i;`
5. `i++`
6. `i < 3;`
7. `somme = somme + i;`

# Déroulement des instructions

```
int i, somme;  
  
somme = 0;  
for (i = 1; i < 3; i++)  
{  
    somme = somme + i;  
}  
write_int(i);
```

## Suite d'Instructions Exécutées

1. somme = 0;
2. i = 1;
3. i < 3;
4. somme = somme + i;
5. i++
6. i < 3;
7. somme = somme + i;
8. i++

# Déroulement des instructions

```
int i, somme;  
  
somme = 0;  
for (i = 1; i < 3; i++)  
{  
    somme = somme + i;  
}  
write_int(i);
```

## Suite d'Instructions Exécutées

1. somme = 0;
2. i = 1;
3. i < 3;
4. somme = somme + i;
5. i++
6. i < 3;
7. somme = somme + i;
8. i++
9. i < 3;



# Déroulement des instructions

```
int i, somme;  
  
somme = 0;  
for (i = 1; i < 3; i++)  
{  
    somme = somme + i;  
}  
write_int(i);
```

## Suite d'Instructions Exécutées

1. somme = 0;
2. i = 1;
3. i < 3;
4. somme = somme + i;
5. i++
6. i < 3;
7. somme = somme + i;
8. i++
9. i < 3;
10. write\_int(i);

## Example

```
POINT p1,p2;

init_graphics(600,300);
p1.y = 10;
p2.y = 290;
for (p1.x = 10; p1.x < 590 ; p1.x = p1.x + 40)
{
    p2.x = p1.x;
    draw_line(p1,p2,bleu);
}
wait_escape();
exit(0);
```

## Exemple : on compte en décrémentant

```
int i;  
  
for(i = 10; i > 0;i--)  
{  
    write_int(i);  
}
```

# Attention Piège !

## Boucles infinies

```
for(i = 0; i < 10; var++)  
{  
    write_int(var);  
}
```

# Attention Piège !

## Boucles infinies

```
for(i = 0; i < 10; var++)  
{  
    write_int(var);  
}
```

```
for(i = 7; i < 12; i--)  
{  
    write_int(i);  
}
```

# Attention Piège !

## Boucles infinies

```
for(i = 0; i < 10; var++)  
{  
    write_int(var);  
}
```

```
for(i = 7; i < 12; i--)  
{  
    write_int(i);  
}
```

## Boucles sans effet

```
a = 6;  
for(i = 2; i*i > a; i = i + 4)  
{  
    write_int(i);  
}
```

# Attention Piège !

## Boucles infinies

```
for(i = 0; i < 10; var++)  
{  
    write_int(var);  
}
```

```
for(i = 7; i < 12; i--)  
{  
    write_int(i);  
}
```

## Boucles sans effet

```
a = 6;  
for(i = 2; i*i > a; i = i + 4)  
{  
    write_int(i);  
}
```

```
for(i = 7; i > 0; i--);  
{  
    write_int(i);  
}
```

# Equivalence entre le while et le for

```
for(initialisation;condition;poursuite)
{
    bloc d'instructions
}
```

---

```
initialisation;
while(condition)
{
    bloc d'instructions
    poursuite;
}
```



do .... while

- Faire un *bloc d'instructions* et ensuite le répéter tant qu'une *condition* est vraie.

```
do  
{  
    bloc d'instructions  
} while(condition);
```

# Déroulement des instructions

```
int i;  
do  
{  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
} while (i <= 0);  
write_text("Merci!");
```

Suite d'Instructions Exécutées

# Déroulement des instructions

```
int i;  
do  
{  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
} while (i <= 0);  
write_text("Merci !");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();

i = -12

# Déroulement des instructions

```
int i;  
do  
{  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
} while (i <= 0);  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();
2. while (i <= 0)

i = -12

# Déroulement des instructions

```
int i;  
do  
{  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
} while (i <= 0);  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();
2. while (i <= 0)
3. write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();

i = 56

# Déroulement des instructions

```
int i;  
do  
{  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
} while (i <= 0);  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();
2. while (i <= 0)
3. write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();
4. while (i <= 0)

i = 56

# Déroulement des instructions

```
int i;  
do  
{  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
} while (i <= 0);  
write_text("Merci!");
```

**i = 56**

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();
2. while (i <= 0)
3. write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();
4. while (i <= 0)
5. write\_text("Merci!");

# Déroulement des instructions

```
int i;  
do  
{  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
} while (i <= 0);  
write_text("Merci!");
```

Suite d'Instructions Exécutées



# Déroulement des instructions

```
int i;  
do  
{  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
} while (i <= 0);  
write_text("Merci !");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();

i = 6367

# Déroulement des instructions

```
int i;  
do  
{  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
} while (i <= 0);  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();
2. while (i <= 0)

i = 6367

# Déroulement des instructions

```
int i;  
do  
{  
    write_text("Entrez un entier > 0");  
    i = lire_entier_clavier();  
} while (i <= 0);  
write_text("Merci!");
```

## Suite d'Instructions Exécutées

1. write\_text("Entrez un entier > 0");  
i = lire\_entier\_clavier();
2. while (i <= 0)
3. write\_text("Merci!");

i = 6367

# Example

```
POINT p1,p2;

init_graphics(600,300);
p1.y = 10;
p2.y = 290;
p1.x = 10;
do
{
    p2.x = p1.x;
    draw_line(p1,p2,bleu);
    p1.x = p1.x + 40;
} while (p1.x < 590);
wait_escape();
exit(0);
```

# Example

```
POINT p1,p2;
int i;
init_graphics(600,300);
i = lire_entier_clavier();
p1.x = 10;
p2.x = 590;
if (i%2 == 0)
{
    for(p1.y = 10; p1.y < 290; p1.y = p1.y + 10)
    {
        p2.y = p1.y;
        draw_line(p1,p2,bleu);
    }
}
else
{
    for(p1.y = 10; p1.y < 290; p1.y = p1.y + 5)
    {
        p2.y = p1.y;
        draw_line(p1,p2,rouge);
    }
}
```

# Example

```
POINT p1,p2;
int i;
init_graphics(600,300);
p1.x = 10;
p2.x = 590;
p1.y = 10;
for(i = 0; i < 10; i = i+1)
{
    p2.y = p1.y;
    if (i%2 == 0)
    {
        draw_line(p1,p2,vert);
    }
    else
    {
        draw_line(p1,p2,bleu);
    }
    p1.y = p1.y + 30;
}
```

# Exemple

```
POINT p1;  
init_graphics(600,300);  
  
for(p1.y = 10; p1.y < 290; p1.y = p1.y + 30)  
{  
    for(p1.x = 10; p1.x < 290; p1.x = p1.x + 30)  
    {  
        if (p1.x == p1.y)  
        {  
            draw_fill_circle(p1,10,rouge);  
        }  
    }  
}
```

# Example

```
POINT p1;
init_graphics(600,300);
for(p1.y = 290; p1.y >= 10; p1.y = p1.y - 30)
{
    for(p1.x = 10; p1.x < 290; p1.x = p1.x + 30)
    {
        if (p1.x == (300 - p1.y))
        {
            draw_fill_circle(p1,10,rouge);
        }
    }
}
```



# Example

```
POINT p1;
init_graphics(600,300);
for(p1.y = 290; p1.y >= 10; p1.y = p1.y - 20)
{
    for(p1.x = 10; p1.x <= 290; p1.x = p1.x + 20)
    {
        if (p1.x == (300 - p1.y))
        {
            draw_fill_circle(p1,10,rouge);
        }
        if (p1.x == p1.y)
        {
            draw_fill_circle(p1,13,vert);
        }
    }
}
```

# Conclusion

Les boucles permettent de répéter des instructions :

- ▶ le corps de la boucle est exécutée au moins une fois (`do` .... `while()` ;).
- ▶ le corps de la boucle peut ne pas être exécuté (`for()` ; ; ) et `while`).