

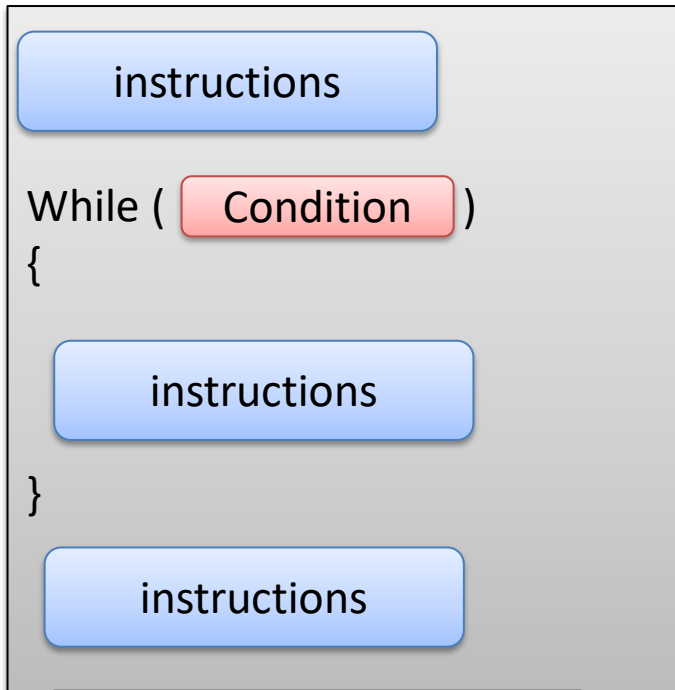
La Programmation en langage C

Partie 2 : les instructions de contrôle

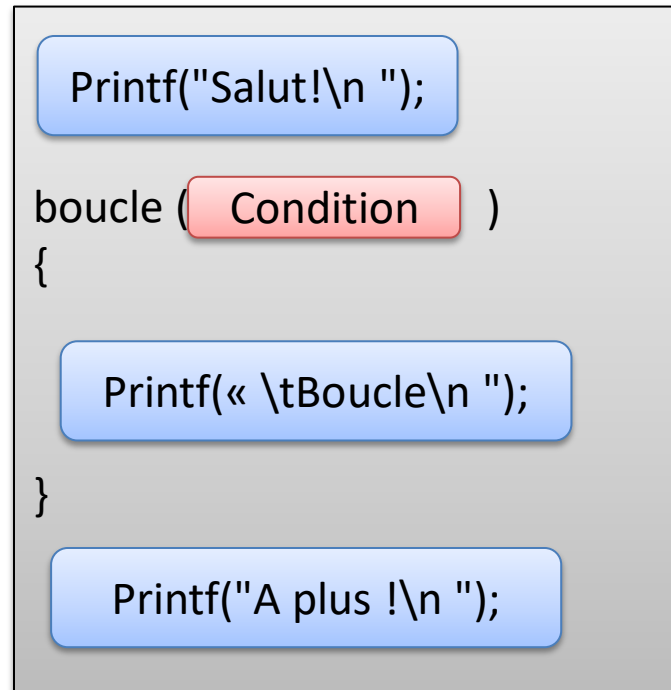
La cible : SMI S3

Pr S.ELFILALI

Les boucles



Structure



Exemple

Salut !

Boucle
Boucle
Boucle
Boucle
Boucle

A plus !

Boucles en C : while, do while et for

Les boucles

C'est quoi une boucle ?

Boucle c'est un ensemble d'instructions qui est exécuté un certain nombre de fois tant que la condition liée est vraie

Les boucles

La boucle while

```
While ( Condition )  
{  
    instructions  
}
```

Structure

```
Int var =0;  
While ( var < 5 )  
{  
    Printf("%d\n" , var);  
    Var++;  
}
```

Exemple

0
1
2
3
4

Les boucles

La boucle while

Exemple : compte à rebours

Passer sur codeblocks et faire un programme sans boucle et avec la boucle

Utiliser les fonctions sleep et générer le bip par `printf("\a");`

la fonction `Sleep("temps en millisecondes");`, permet de faire attendre le programme pendant un certains nombre de millisecondes avant de continuer à exécuter les instructions.

```
#include <stdio.h>
#include <stdlib.h>
```

```
int main()
{
    int compte=10;
    printf("\a"); // bip
    getch();
    printf("explosion dans 10 s");

    printf("\ncompte à rebours ..... !\n");

    printf("\n%d",compte);
    compte--;
    sleep(1); // attente en millisecondes

    printf("\n%d",compte);
    compte--;
    sleep(1);

    printf("\n%d",compte);
    compte--;
    sleep(1);

    .....

    printf("\n%d",compte);
    compte--;
    sleep(1);

    printf("\n%d",compte);
    compte--;
    sleep(1);
    printf("\a");

    printf("\nBOOOOOOMMM");
    return 0;
}
```

La boucle while

La boucle while

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int compte=10;
    printf("\a");
    getch();
    printf("explosion dans 10 s");

    printf("\ncompte à rebours ..... !\n");

    while (compte>0)
    {
        printf("\n%d",compte);
        compte--;
        sleep(1);
    }

    printf("\a");

    printf("\nBOOOOOO MMM");
    return 0;
}
```

Les boucles

La boucle do while

```
do  
{  
    instructions  
} While ( Condition );
```

Structure

```
Int var =6;  
do  
{  
    Printf("%d\n" , var);  
    Var++;  
} While ( var < 5 );
```

Exemple

6

Les boucles

La boucle do while

Exemple : contrôle de saisie

Passer sur codeblocks et faire un programme qui demande à l'utilisateur de saisir un nombre entre 1 et 10

Les boucles

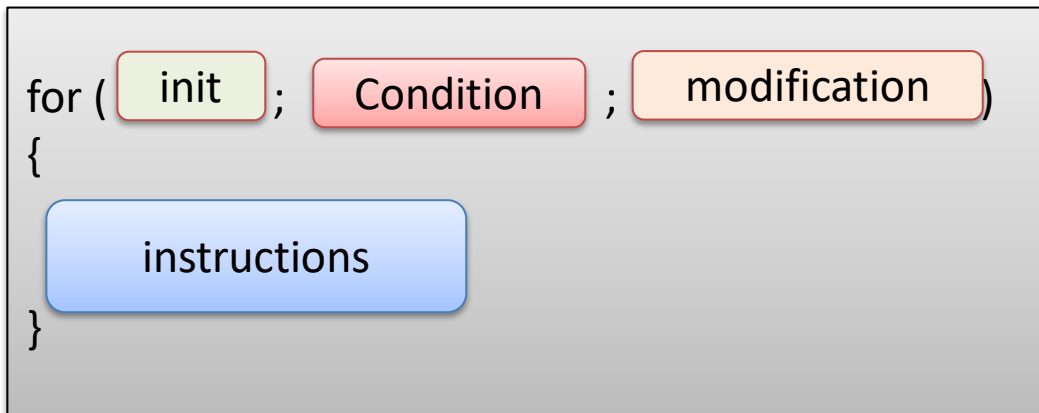
La boucle do while

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int saisie_utilisateur = 0;
    do
    {
        printf("donner un nombre entre 1 et 10 : ");
        scanf("%d",&saisie_utilisateur);
    }
    while (saisie_utilisateur <1 || saisie_utilisateur >10);
    return 0;
}
```

Les boucles

La boucle for



Structure

```
for (int var =0 ; var<5 ; var++ )  
{  
    printf("%d\n",var);  
}
```

Exemple

0
1
2
3
4

```
int var =0 ;  
while(var<5)  
{  
    printf("%d\n",var);  
    var ++ ;  
}
```

Equivalent

0
1
2
3
4

Les boucles

La boucle for

Exemple : compte à rebours

Passer sur codeblocks et faire un programme avec la boucle for

Utiliser les fonctions sleep et générer le bip par `printf("\a");`

la fonction `Sleep("temps en millisecondes");`, permet de faire attendre le programme pendant un certains nombre de millisecondes avant de continuer à exécuter les instructions.

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int compteur = 10;
    printf("Explosion dans  ");
    for( compteur = 10 ; compteur>0 ; compteur--)
    {
        printf("\n%d", compteur);
    }
    printf("\n\na BOOM");
    return 0;
}
```

Règles d'or des boucles

Comment choisir la boucle qu'on va utiliser ?

Si on connaît le nombre d'itération du bloc d'instruction alors on utilise la boucle for

Si on connaît pas le nombre d'itération du bloc alors nous avons deux choix :

Si on veut exécuter au moins une fois le bloc d'instruction alors c'est la boucle « do .. While »

Et dans tous les autres cas , on utilise la boucle while

Règles d'or des boucles

Boucle infinie

C'est une boucle qui va s'exécuter à l'infini!

Cad que sa condition ne va jamais passer à faux!!
Le programme va se bloquer !!

Le seul moyen pour le débloquent : **Ctrl+C** ou **Alt F4**

Exemple

Apportez les modification suivantes au programme et commentez :

1. Compteur ++
2. Modifier le type en short et compteur ++
3. Modifier la condition : >-99999 et compteur --

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int compteur = 10;
    printf("Explosion dans ");
    for( compteur = 10 ; compteur>0 ; compteur --)
    {
        printf("\n%d", compteur);
    }
    printf("\n\na BOOM");
    return 0;
}
```

Règles d'or des boucles

Boucle infinie

Apportez les modifications suivantes au programme et commentez :

1. **Compteur ++** : boucle infinie, on n'arrive pas à voir le max de ce type, on doit l'arrêter par Ctrl+C ... la condition peut être fautive au max de ce type
2. **Modifier le type en short et compteur ++** : il s'arrête au max du short qui est 32767 et puis il passe à la première valeur de type int avec une incrémentation et dans ce cas la condition n'est pas vérifiée et la boucle s'arrête !
3. **Modifier la condition : >-99999 et compteur --** : la condition est toujours vérifiée et il ne va jamais s'arrêter !!!!!

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int compteur = 10;
    printf("Explosion dans ");
    for( compteur = 10 ; compteur>0 ; compteur --)
    {
        printf("\n%d", compteur);
    }
    printf("\n\n BOOM");
    return 0;
}
```


Les boucles

Exercice : code PIN

- 1- Créer un programme qui demande à l'utilisateur de rentrer le code PIN à 4 chiffres. Si il trouve le bon code, afficher le message : téléphone déverrouillé sinon afficher un message d'erreur et redemander d'entrer le code.
- 2- Gérer un nombre maximum de tentatives: Exemple 3 tentatives

```
Code PIN: 1234
  Erreur, il vous reste 2 tentatives.
Code PIN: 0000
  Telephone deverrouille
```

- 1- On ne connaît pas le nombre d'itération mais on souhaite entrer dans la boucle au moins une fois. On utilise alors une boucle "Do While"
- 2- Créer une constante pour le nombre maximum de tentatives.

Les boucles

Solution : code PIN

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    const int CODE_PIN = 0000;
    const int MAX_TENTATIVE = 3;
    int saisie;
    int nbre_Tentative= MAX_
TENTATIVE;
    // lecture de la saisie

    do {
        printf("Code Pin : ");
        scanf("%d",&saisie);

        if(saisie==CODE_PIN)
        {
            printf("\tTelephone deverrouille\n");
        }
        else
        {
            nbre_Tentative --;
            printf("\tErreur , il vous reste %d tentatives\n", nbre_Tentative);
        }
    }
    while(saisie!=CODE_PIN && nbre_Tentative > 0 );
    return 0;
}
```

Code PIN: 1234
Erreur, il vous reste 2 tentatives.
Code PIN: 0000
Telephone deverrouille

Les boucles

Break et Continue

break : sortir de la boucle courante

```
int main()
{
    int var ;
    printf("debut \n");
    for (var =0 ; var<5 ; var++ )
    {
        if(var ==2)
        {
            break;
        }
        printf("%d\n",var);
    }
    printf("fin \n");
    return 0;
}
```

debut
0
1
fin

break

continue : sauter a la fin de la boucle

```
int main()
{
    int var ;
    printf("debut \n");
    for (var =0 ; var<5 ; var++ )
    {
        if(var ==2)
        {
            continue;
        }
        printf("%d\n",var);
    }
    printf("fin \n");
    return 0;
}
```

debut
0
1
3
4
fin

continue

Les boucles

Break et Continue : exemple

```
int main()
{
    const int CODE_PIN = 0000;
    const int MAX_TENTATIVE = 3;
    int saisie_utilisateur;
    int nbre_Tentative= 0;
    int compteur;
    for( nbre_Tentative= 1 ;nbre_Tentative<=MAX_TENTATIVE ; nbre_Tentative++)
    {
        printf("Code Pin : ");
        scanf("%d",&saisie_utilisateur);
        if(saisie_utilisateur<0)
        {
            printf("code utilisateur doit etre superieur a 0 \n");
            continue;
        }
        if(saisie_utilisateur==CODE_PIN)
        {
            printf("\t Bienvenue\n");
            break;
        }
        else
            printf("\t Code incorrect  \n");
    }
    return 0;
}
```

Les boucles

Les boucles imbriquées

Exercice : afficher toutes les tables de multiplication de 1 à 10

Aide : commencer par afficher une seule table de multiplication

Table de 1
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10

Table de 2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20

Table de 3
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30

Table de 4
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
4 x 6 = 24
4 x 7 = 28
4 x 8 = 32
4 x 9 = 36
4 x 10 = 40

Table de 5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50

Table de 6
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60

Table de 7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70

Table de 8
8 x 1 = 8
8 x 2 = 16
8 x 3 = 24
8 x 4 = 32
8 x 5 = 40
8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
8 x 10 = 80

Table de 9
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90

Table de 10
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100

Les boucles

Les boucles imbriquées

Exercice : afficher toutes les tables de multiplication de 1 à 10

Aide : commencer par afficher une seule table de multiplication

```
int main()
{
    int ligne = 0;
    int id_table = 1;

    for(id_table=1; id_table<=10; id_table++)
    {

        printf("table multiplication %d\n", id_table);
        printf("*****\n");
        for(ligne=1; ligne<=10; ligne++)
        {
            printf("\t%d x %d = %d\n", ligne, id_table, ligne*id_table);
        }
        printf("\n") ;
    }
    return 0;
}
```

Les boucles

Jeux kahoot

<https://create.kahoot.it/details/boucles/db371927-a176-4c1f-8811-2b3b73759cdb>

Kahoot!

Les boucles

TP : un jeu

Nous allons créer un petit jeu de type “Le juste prix”. Pour les plus jeunes, cela correspond à un ancien jeu télévisé qui consiste à demander, aux participants, de trouver le prix d'un objet. Ils font alors des propositions et le présentateur leur dit simplement "c'est plus" ou "c'est moins", jusqu'à ce que l'un d'entre eux trouve le bon prix. Mais biensure, il y a une limite de temps! Nous allons donc demander à l'utilisateur de trouver un nombre entier compris entre 1 et 100.

Il vas alors pouvoir faire des propositions et pour chacune d'entre elles, nous allons afficher “c’est plus !” ou “c’est moins !”. lorsque l'utilisateur a trouvé le bon nombre alors on affiche un message de félicitation et on termine le programme. Le joueur a le droit à 10 tentatives.

```
...  
> 23  
c'est plus !  
> 30  
c'est moin !  
> 25  
Bravo! le nombre mystère est bien 25
```


Les boucles

TP : un jeu

Préparatifs

- 1- Créer un nouveau projet C du nom de "tp_les_boucles".
- 2- Ajouter le code suivant en début de fonction main. Celui-ci permet de générer aléatoirement un nombre compris entre 1 et 100.
Attention, il faut inclure time.h pour l'utiliser.

```
// declaration de nos variables
int nombre_mystere = 0, nombre_utilisateur = 0;
const int VALEUR_MIN = 1, VALEUR_MAX = 100;
// generation d'un nombre aleatoire
srand(time(NULL));
nombre_mystere = (rand() % (VALEUR_MAX - VALEUR_MIN + 1)) + VALEUR_MIN;
printf("Devinez quel est mon nombre mystere. Indice: c'est un nombre entre %d et %d
\n", VALEUR_MIN, VALEUR_MAX);
```

Les boucles

TP : un jeu

Instructions

1- Créer la boucle de jeu

1.1 Demander à l'utilisateur de rentrer un entier pour trouver le nombre mystère.

1.2 Comparer la valeur saisie avec le nombre a trouver.

1.3 Gérer la fin de partie (le joueur a trouvé le bon nombre).

2- Ajouter un compteur qui indique combien de tentative a fait le joueur.

...

tentative 3 > 23

c'est plus !

tentative 4 > 30

c'est moin !

tentative 5 > 25

Bravo! le nombre mystère est bien 25

Les boucles

TP : un jeu

Pour aller plus loin

A- Modifier le code pour rendre le jeu plus difficile. pour cela, ajouter un nombre maximum de tentative.

B- Ajouter un menu à l'utilisateur en fin de partie lui permettant de faire une nouvelle partie ou quitter.

```
...
tentative 3/10 > 23
c'est plus !
tentative 4/10 > 30
c'est moin !
tentative 5/10 > 25
Bravo! le nombre mystère est bien 25

Voulez-vous faire une nouvelle partie (1-oui, 2-non)
> 2
A bientôt !
```

Les boucles

TP : un jeu

De l'Aide

1. Faire une boucle avec pour condition le fait que la saisie utilisateur corresponde ou non au nombre à trouver.
2. Il faut créer une nouvelle variable initialisé a 0 et l'incrémenter à chaque tentative de l'utilisateur.

A- Créer une nouvelle constante qui comporte le nombre maximum de tentative. quand ce nombre est atteint, utiliser une des instruction suivante pour terminer la partie (break, continue).

B- Mettre la boucle de jeu dans une nouvelle boucle. la condition de cette nouvelle boucle doit vérifier le choix de l'utilisateur.

Prochain chapitre :
les pointeurs