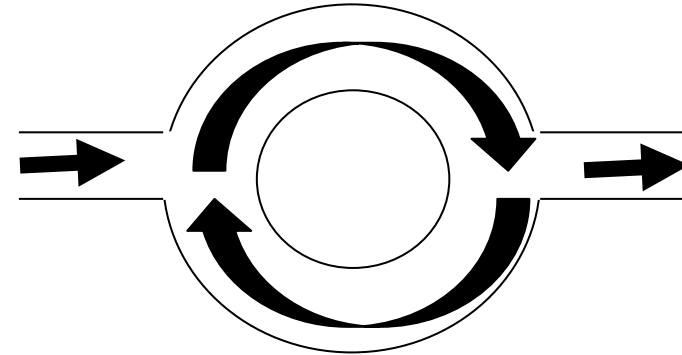


for

while



do-while

Comment répéter une séquence d'instructions ?
Une, deux, ..., N fois?

Langage C

LES BOUCLES

Introduction

Pour tous les amis dont c'est l'anniversaire,
Gaston :

cuisine un gâteau.

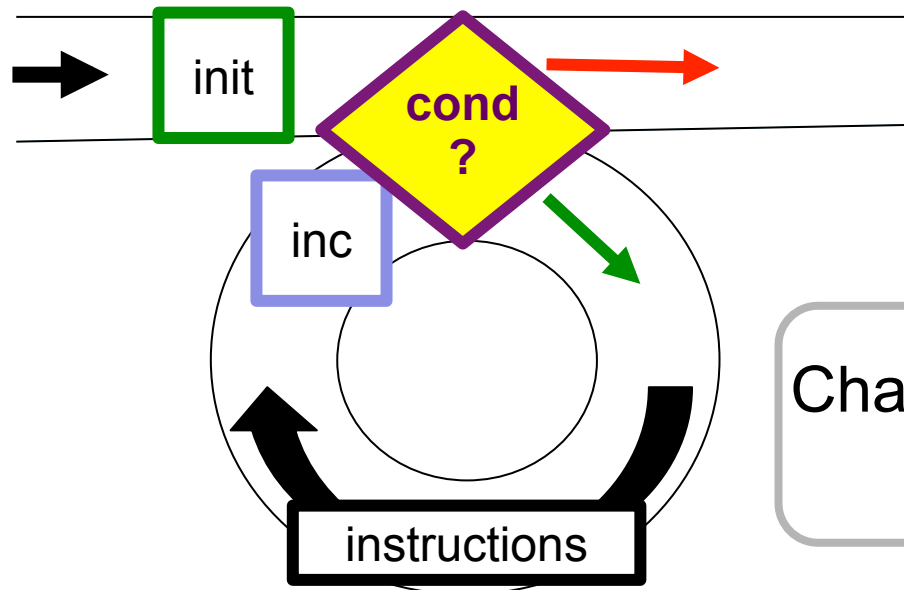
ajoute une carte d'anniversaire

Il faut **répéter** un certain nombre
de fois les instructions à exécuter.



La boucle for

```
for ( <init> ; <condition> ; <inc> )  
{  
  <instructions>  
}
```



<init> et <inc> sont des instructions.

Chaque passage dans la boucle s'appelle une **itération**.

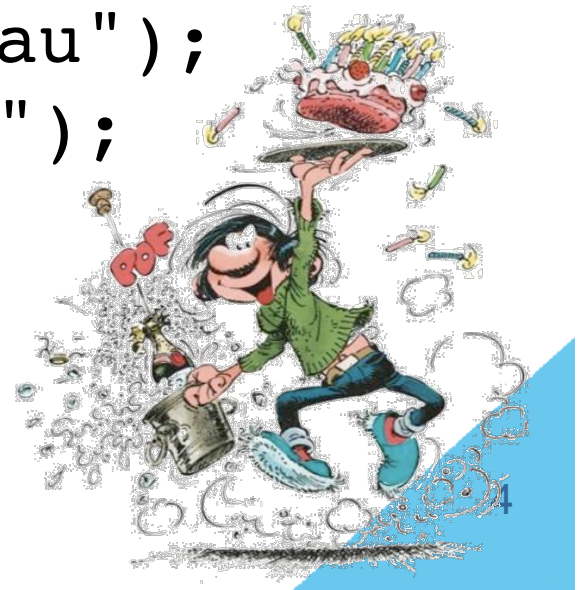
Exemple

i sert ici de compteur d'itération

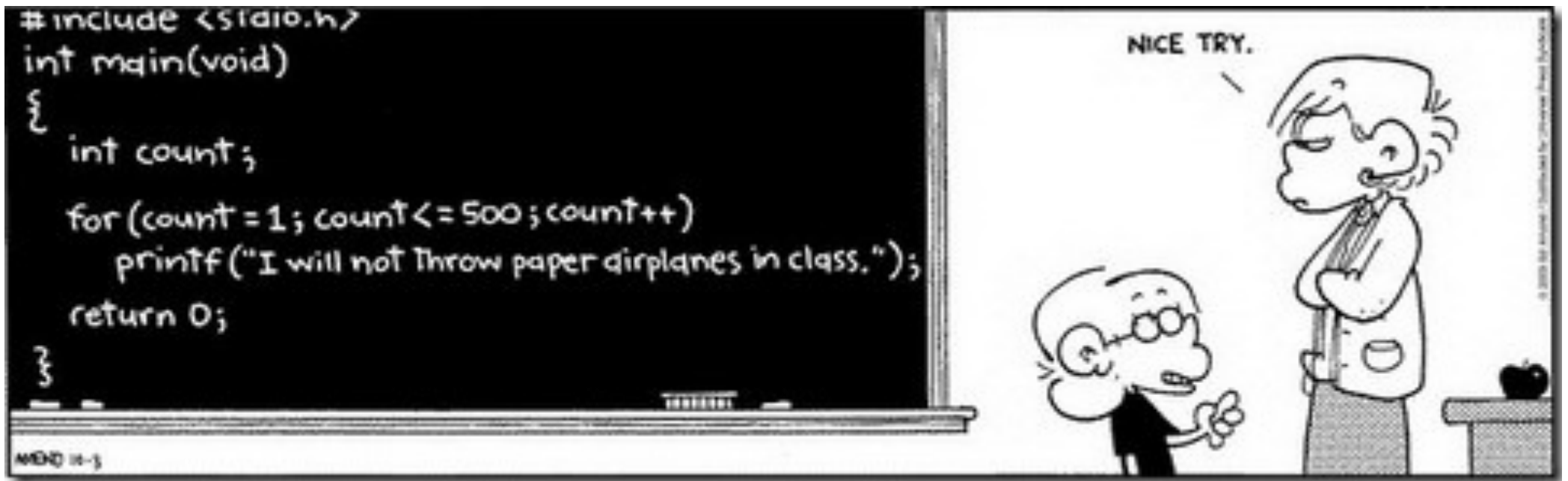
```
int n;  
printf("Nombre de cadeaux à faire :");  
scanf("%d", nbCad);  
int i;  
for ( i = 1 ; i <= nbCad ; i = i+1 )  
{  
    printf("Gaston cuisine un gâteau");  
    printf("Gaston écrit une carte");  
}
```

Alternative classique :

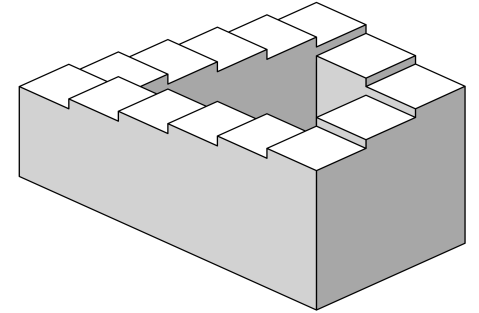
```
for (int i = 0 ; i < nbCad ; i ++ ) { ... }
```



...



Attention aux boucles infinies



```
for (i = 0 ; i < 10 ; i--) {  
    printf("X");  
}
```



ATTENTION !

Une boucle infinie est en général une erreur de programmation.

Exercice dixX



Le programme affiche une séquence de dix X.

```
> ./dixX  
XXXXXXXXXX
```

```
for ( int i = 0 ; i < 10 ; i = i + 1 ) {  
    printf( "X" );  
}
```

Exercice unadix



Le programme affiche une séquence les nombres de 1 à 10
séparés par des tirets.

```
> ./unadix  
1-2-3-4-5-6-7-8-9-10
```

```
for (int i = 1 ; i <= 10 ; i++) {  
    // Pas de tiret pour le premier nombre  
    if (i!=1) printf("-");  
    // Affiche le compteur  
    printf("%d", i);  
}
```


Exercice minuscinq



Le programme affiche une suite décroissante de 5 en 5.

```
> ./minuscinq  
50, 45, 40, 35, 30, 25.
```

Exercice minuscinq – solution 1



```
for (int i = 50; i >= 25; i = i-5) {  
    // Préfixe sauf pour le premier terme  
    if (i != 50)  
        printf(", ");  
  
    // Affiche le nombre (la valeur du compteur)  
    printf("%d", i);  
}  
  
printf(".\n");
```

Exercice minuscinq – solution 2



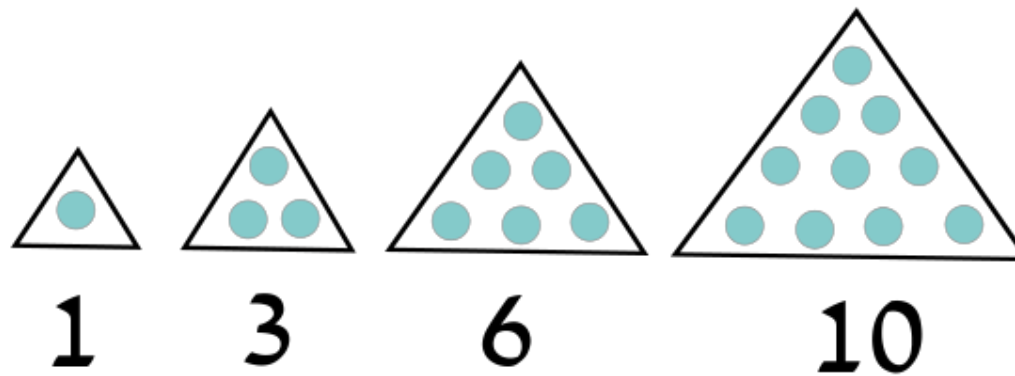
```
for (int i = 0; i < 6; i++) {  
  
    // Préfixe sauf pour le premier terme  
    if (i != 0)  
        printf(", ");  
  
    // Affiche le nombre (calculé)  
    printf("%d", 5 * (10-i));  
}  
  
printf(".\n");
```

Exercice nbtriang



Le programme affiche les 5 premiers nombres triangulaires.
Le premier est 1, puis chacun est calculé en ajoutant son rang au précédent.

```
> ./nbtriang  
1 3 6 10 15
```



Rang → 1 2 3 4

Exercice nbtriang – Solution 1

```
// Contiendra le nombre triangulaire
int nbtriang = 0;

for (int rang = 1; rang <= 5; rang++) {

    // Tilet en préfixe sauf pour le premier nombre
    if (rang != 1)
        printf(" ");

    nbtriang = nbtriang + rang;
    // Affiche le nombre triangulaire pour l'itération
    printf("%d", nbtriang);
}
printf("\n");
```

Exercice nbtriang – Solution 2

```
// le nombre triangulaire de l'itération courante
int nbtriang = 0;

for (int rang = 1; rang <= 5; rang++) {

    // Affiche espace en préfixe (sauf 1er nombre)
    if (rang != 1) printf(" ");

    // Nombre triangulaire est calculé comme somme
    // de son rang et du nombre triangulaire précédent
    nbtriang = nbtriang + rang;

    // Affiche le nombre ()
    printf("%d", nbtriang);

}
```

Exercice carrelette



Le programme affiche un carré composé d'un caractère.

```
> ./carrelette  
Dimension du carré : 3  
XXX  
XXX  
XXX
```

Solution carrelette



```
// dimension du carré
int dim;

// Saisies
printf("Dimension du carré :");
scanf("%d", &dim);

// Afficher nb lignes..
for (int lgn = 0; lgn < dim; lgn++) {
    // ... de nb caractères
    for (int col = 0; col < nb; col++) {
        printf("X");
    }
    // Fin de ligne
    printf("\n");
}
```


Exercice rectanglelettre



Le programme affiche un rectangle composé d'un caractère.

```
> ./rectanglelettre
```

```
Largeur : 4
```

```
Hauteur : 3
```

```
XXXX
```

```
XXXX
```

```
XXXX
```

Solution rectanglelettre



```
// Dimensions du rectangle
int largeur, hauteur;

printf("Largeur :");
scanf("%d", &largeur);
printf("Hauteur :");
scanf("%d", &hauteur);

for (int lgn = 0; lgn < hauteur; lgn++) {
    for (int col = 0; col < largeur; col++)
        printf("X");
    printf("\n");
}
```

Exercice trianglehaut



L'utilisateur saisit la hauteur du triangle. Le programme affiche un triangle composé d'étoiles. Le programme sort si la hauteur n'est pas valide.

```
> ./trianglehaut
Hauteur [1-10] : 4
*
**
***
****

> ./trianglehaut
Hauteur [1-10] : 0
Désolé, hauteur non valide.
```

Solution trianglehaut

```
int hauteur;

printf("\nHauteur [1-10] : ");
scanf("%d", &hauteur);

if (hauteur < 1 || hauteur > 10) {
    printf("Désolé, hauteur non valide.");
    return EXIT_FAILURE;
}

// Afficher <hauteur> lignes..
for (int lgn = 1; lgn <= hauteur; lgn++) {

    // ... de <lgn> colonnes
    for (int col = 0; col < lgn; col++)
        printf("*");

    printf("\n");
}
```


Factorielle - Solution 1

```
int n;  
int factorielle = 1;  
  
printf("Nombre : ");  
scanf("%d", &n);  
  
// On commence à 2 car 1*1 = 1,  
// donc la première itération est inutile  
for(int i = 2 ; i <= n ; i++)  
    factorielle *= i;  
  
printf("%d! = %d" , n, factorielle);
```

Exercice Factorielle (2)



Quel souci avec cette solution? Proposez une amélioration.

Factorielle – Solution 2

```
int n;  
unsigned long long int factorielle = 1;  
  
printf("Nombre : ");  
scanf("%d", &n);  
  
// On commence à 2 car 1*1 = 1,  
// donc la première itération est inutile  
for(int i = 2 ; i <= n ; i++)  
    factorielle *= i;  
  
printf("%d! = %llu" , n, factorielle);
```


Exercices complémentaires



```
trianglebas(3)
```

```
***  
**  
*
```

```
pyramide(3)
```

```
  *  
 ***  
*****
```

```
sapin(3, 3)
```

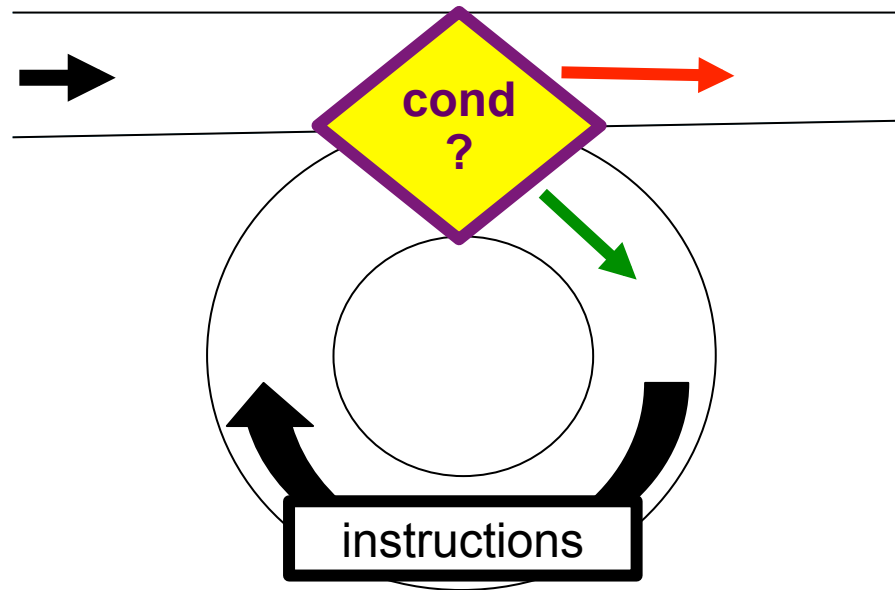
```
  *  
 ***  
*****  
 ***  
*****  
*****  
*****  
*****  
*****  
*****  
*
```

Langage C

LA BOUCLE WHILE () {}

Syntaxe de while

```
while ( <condition> ) {  
    <instructions>  
}
```



Exemples

```
int i = 0;
while (i < 10) {
    printf("X");
    i = i + 1;
}
```

XXXXXXXXXX

Le nombre d'itérations est connu à l'avance, on préférera la solution avec **for**

```
int n = 1;
while (n * (n+1) < 200) {
    n = n + 1;
}
printf(n);
```

Plus petite valeur n telle que $n(n+1)$ dépasse 200.

Le nombre d'itérations n'est pas connu à l'avance, on préférera cette solution

A vous de jouer



Plus petite valeur n telle que $1+2+\dots+n$ dépasse 200.

```
int n = 1, cumul = 0;
while (cumul < 200) {
    n = n + 1;
    cumul = cumul + n;
}
printf(n);
```

A vous de jouer!



Ecrire un programme **copychar**

- l'utilisateur écrit au clavier
- Après l'appui sur la touche entrée, le programme recopie ce que l'utilisateur a entré.
- Le programme s'arrête quand l'utilisateur entre la fin de fichier (CTRL-D)

`c = getchar();` // c contiendra le caractère suivant du flot d'entrée

`putchar(c)` affiche le contenu de la variable entière `c` sous forme d'un caractère

La fin des données est indiquée par la valeur entière EOF (`stdio.h`)

Corrigé 1 copychar.c

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

int main(void) {

    int c; // int pour supporter le cas EOF
    c = getchar();
    while (c != EOF) {
        putchar(c);
        c= getchar();
    }

    return EXIT_SUCCESS;
}
```

Corrigé 2 copychar.c

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

int main(void) {

    int c; // int pour supporter le cas EOF
    while ((c = getchar()) != EOF) {
        putchar(c);
    }

    return (EXIT_SUCCESS);
}
```


A vous de jouer!



Ecrire un programme eofval qui affiche la valeur de de la constante EOF sur votre environnement

Corrigé eofval.c

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

int main(void) {

    printf("EOF = %d", EOF);
    return (EXIT_SUCCESS);
}
```

A vous de jouer!



Ecrire un programme countchar

- Le programme laisse l'utilisateur saisir un texte jusqu'à ce qu'il entre le caractère de fin de fichier
- Le programme affiche le nombre de caractères saisis.

Corrigé countchar.c

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

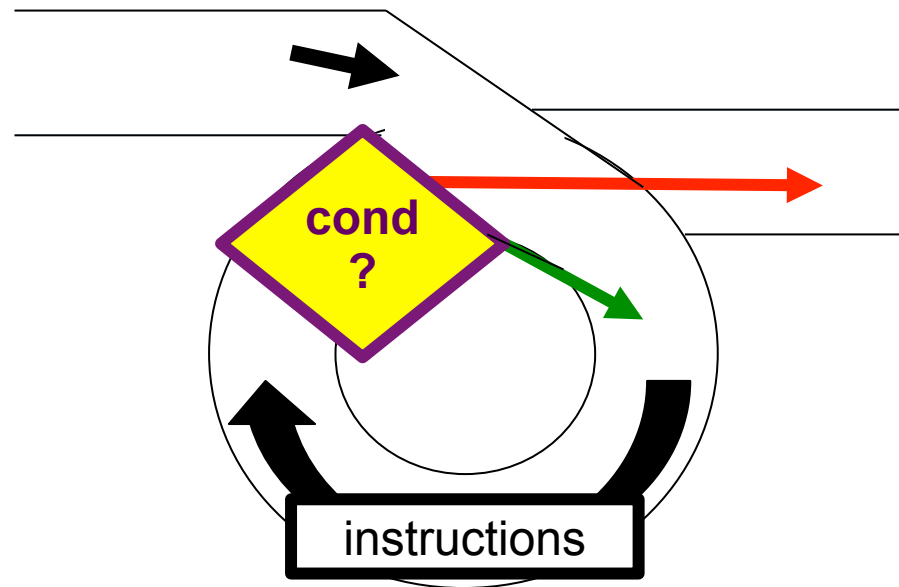
int main(void) {

    int c; // int pour supporter le cas EOF
    int i = 0;
    while ((c = getchar()) != EOF)
        i++;
    printf("%d caractères saisis", i);

    return (EXIT_SUCCESS);
}
```

La boucle do while

```
do {  
    <instructions>  
} while ( <condition> ) ;
```



A vous de jouer



Modifier le programme **trianglehaut** pour que toute valeur non valide soit refusée, et que le programme demande alors de saisir à nouveau le nombre.

```
> ./trianglehaut
```

```
Entrez la hauteur [1-10] : 12
```

```
Désolé, votre saisie est non valide.
```

```
La valeur attendue doit être entre 1 et 10.
```

```
Entrez la hauteur [1-10] : 4
```

```
*
```

```
**
```

```
***
```

```
****
```

Exercice complémentaire



L'utilisateur saisit autant de valeurs qu'il le souhaite.
Quand il saisit 0, le programme s'arrête en affichant le cumul
des saisies.

```
int n, cumul = 0;  
do {  
    printf("\nVotre saisie");  
    scanf("%d", &n);  
    cumul = cumul + n;  
} while (n != 0);  
printf("\nCumul = %d", cumul);
```

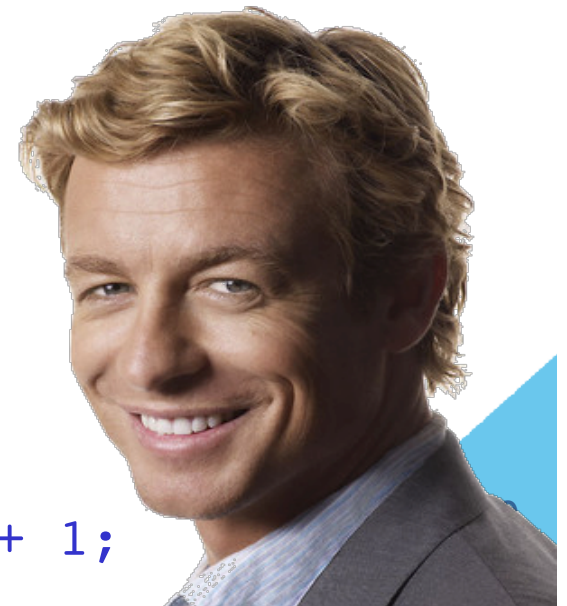
Exercice mentaliste



- Le programme choisit aléatoirement un code secret (un entier entre 1 et 100).
- L'utilisateur dispose de 8 essais pour trouver ce code secret.
- Au bout des 8 essais infructueux, l'utilisateur a perdu et le programme révèle le code secret.
- A chaque essai:
 - L'utilisateur propose un nombre (une saisie hors limite n'est pas prise en compte)
 - S'il s'agit du nombre secret, c'est gagné et le nombre d'essais qui aura été nécessaire est affiché.
 - Sinon le programme affiche si le nombre secret est plus grand ou plus petit que le nombre.

Générer un nombre
pseudo-aléatoire
entre 1 et 100

```
#include <time.h>
#include <stdlib.h>
// ... Dans le code ...
srand(time(NULL));
int codeSecret = rand() % 99 + 1;
```



Exercice retourne



- L'utilisateur saisit un nombre entier
- Le programme affiche ce nombre dans l'ordre inverse de ses chiffres

> **retourne**

Retourneur de nombre saisi.

Entrez le nombre : **12345**

Le nombre retourné est 54321.

Corrigé retourne.c

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

int main(void) {

    int n;
    printf("Retourneur de nombre saisi.");
    printf("\nEntrez le nombre : ");
    scanf("%d", &n);
    int reste;
    while (n > 0) {
        reste = n % 10;
        printf("%d", reste);
        n = n / 10;
    }

    return (EXIT_SUCCESS);
}
```