GROOT ACADEMY

Loops in C

↳ for loop syntax : for (initialization; condition; increment / decrement) {

// code

}

⇒ Example One → Print numbers from 1 to 10.

```
#include <stdio.h>
int main () {
  for ( int i = 1 ; i <= 10 ; i++){
      printf ("%d", i);
  }
  return 0;
}
```

⇒ i starts from 1 and increases each time.

⇒ Example Two → sum of first N natural numbers

```
#include <stdio.h>
int main () {
    int n, sum = 0;
  printf (" enter a number : ");
  scanf ("%d", &n);
  for ( int i = 1; i <= n ; i++) {
        sum += i    (sum = sum + i)
  }
  printf (" sum of first %d natural numbers is
                %d \n", n, sum);
  return 0;
  }
```

+91-8233266276
info@grootacademy.com
122/166, 2nd Floor, vijay path, Mansarovar, 302020
www.Grootacademy.com

→ Example 3 : Print Multiplication Table

```
#include <stdio.h>
int main(){
    int num;
    printf("enter a number");
    scanf("%d", &num);

for(int i=1; i<=10; i++){
    printf("%d x %d = %d \n", num, i, num*i);
    }
return 0;
}
```

→ Example 4 : Reverse Counting from N to 1

```
#include <stdio.h>
int main(){
    int N;
    printf("enter a number:");
    scanf("%d", &N);
for(int i=N; i>=1; i--){
        printf("%d", i);
}
return 0;
}
```

⇒ Example 5 : Print a simple Pyramid System

```
#include <stdio.h>
int main() {
    int rows;
    printf("enter no. of rows: ");
    scanf("%d", &rows);

    for(int i=1; i<=rows; i++) {
        for(int j=1; j<=i; j++) {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

⇒ while Loop :

⇒ Example 1 : Keep tracking user input, until they enter a negative number

```
#include <stdio.h>
int main(){
    int num;
    printf("enter a negative no to stop");
    scanf("%d", &num);

    while(num>=0){
        printf("you entered:");
        scanf("%d", &num);
    }
    printf("loop exited due to neg. input");
    return 0;
}
```

GROOT ACADEMY

⇒ Example 2: find the sum of even numbers until N.

```
#include <stdio.h>
int main(){
    int N, sum=0, i=2;        // 1st even no.
    printf("enter the number:");
    scanf("%d", &N);
    while(i<=N){
        sum+=i;               // add even no. to sum
        i+=2;                 // jump to next even no.
    }
    printf("sum of even numbers: %d", sum);
    return 0;
}
```

⇒ Example 3: Guess the correct number

```
#include <stdio.h>
int main(){
    int secret=7, guess;
    printf("guess any no. b/w 1-10 :");
    scanf("%d", &guess);

    while(secret!=guess)
    while(guess!=secret){
        printf("Wrong, try again");
        scanf("%d", &guess);
    }
    printf("congrats, you've entered correct no.");
    return 0;
}
```

+91-8233266276
info@grootacademy.com
122/166, 2nd Floor, vijay path, Mansarovar, 302020
www.Grootacademy.com

→ Example 4: ATM withdrawal simulation (balance check);

```
#include <stdio.h>
int main () {
    int balance = 10000, withdraw;
while (1) {
    printf ("enter amount to withdraw (or 0 to exit:) ");
    scanf ("%d", &withdraw);
if (withdraw == 0){
    printf ("transaction ended. \n");
    break;
} else if (withdraw > balance) {
    printf ("insufficient funds! Your balance is
        %d.\n" , balance);
} else {
    balance -= withdraw;
    printf ("withdrawal successful! Remaining
        balance : %d \n", balance);
}
}
return 0;
}
```

Note: while(1) creates an infinite loop, the condition inside while must be true for the loop to continue., while(1) means "keep running forever, until you manually stop it using break.

⇒ Key differences (for vs while loop)

- ⊙ use for loop when working with counters. No. of iterations are known.
- ⊙ use while loop when 'waiting for a condition'.
- ⊙ for loop is well structured, while loop can be complex.
- ⊙ For loop = counting loops, iterating over arrays
  while loop = loops based on user input, real time events.

The if-else statments:

⊙ Example (- Check even or odd

```
#include <stdio.h>
    int main () {
        int num;
        printf(" enter the number:");
        scanf("%d", &num);

    if (num % 2 == 0) {
            printf("%d is even \n", num);
        } else {
            printf("%d is odd \n", num);
    }
    return 0;
}
```

→ Example 2- grading system ex-

```
#include <stdio.h>
int main () {
    int marks;
    printf (" enter marks : ");
    scanf (" %d", &marks);

if (marks >= 90) {
    printf (" gradi:A \n");
} else if (marks >= 80) {
    printf (" grade : B \n");
} else if (marks >= 70) {
    printf (" gradi : c \n");
} else if (marks >= 60) {
    printf (" gradi : D \n");
} else {
    printf (" grade : F (fail) \n");
}
return 0;
}
```

→ Example 3- switchCase (simple calculator pgm)

```
#include <stdio.h>
int main () {
    int num1, num2, choice;
    printf (" enter two numbers : ");
    scanf (" %d %d ", &num1, &num2);
    printf (" choose an operation! ");
```

```
printf ("1. addition \n 2. subtraction \n 3. Multiplication \n
        4. Division");
scanf ("%d", &choice);

switch (choice) {
    case 1:
        printf ("Result : %d \n", num1 + num2);
        break;
    case 2:
        printf ("Result : %d \n", num1 - num2);
        break;
    case 3:
        printf ("Result : %d \n", num1 * num2);
        break;
    case 4:
        if (num2 != 0) {
            printf ("Result : %.2f \n", (float) num1 / num2);
        else
            printf ("error ! division by zero !");
        break;
    default:
        printf ("invalid choice ! please enter (-4 . \n");
    }
    return 0;
}
```

* NOH: → use switch case when → multiple conditions depend on a single variable