Programming Using C
Week 04-2

Looping
for loop

Prepared By: REC Faculty 4.0 Team

LOOPING STATEMENTS

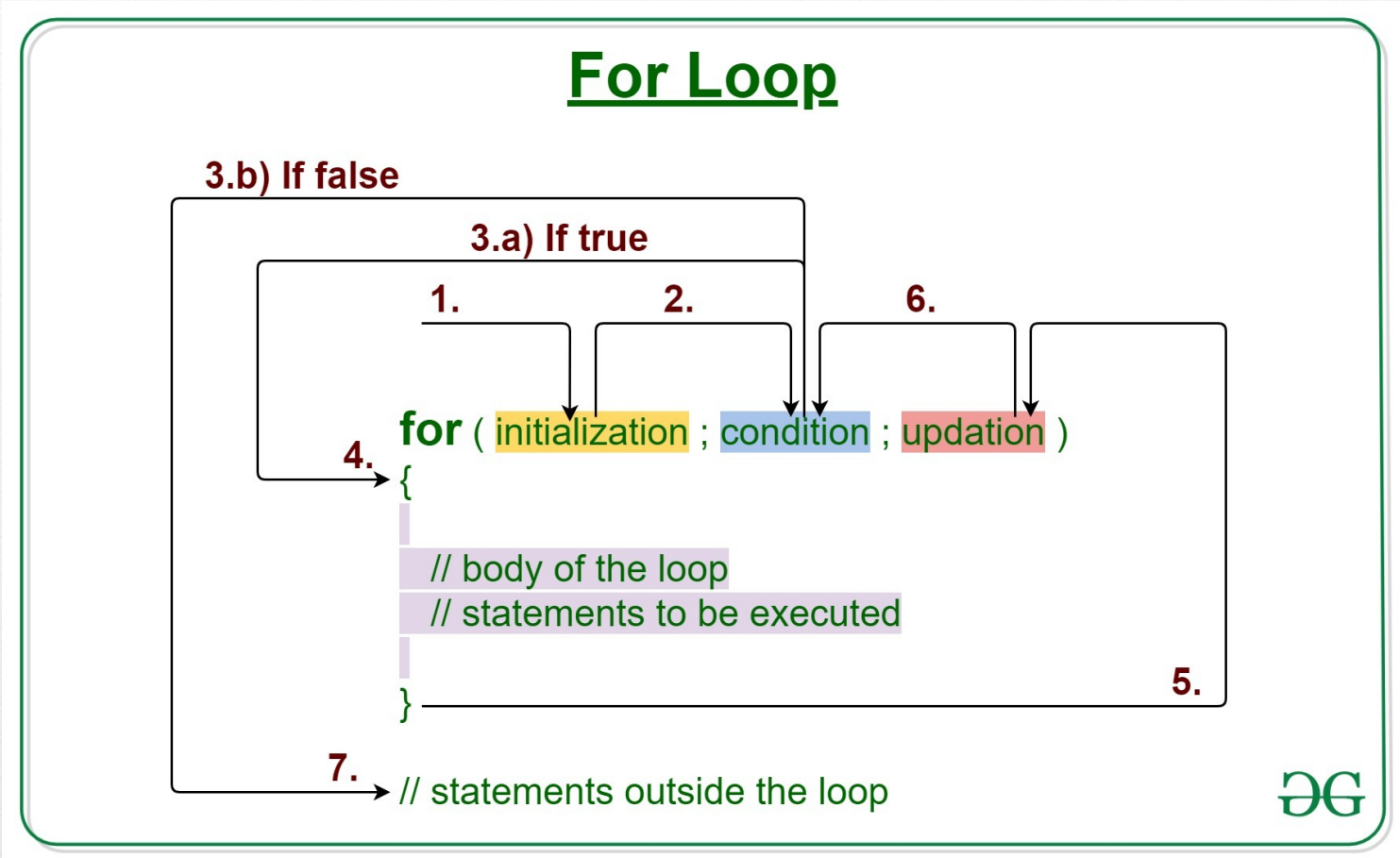
- Loops in programming come into use when we need to repeatedly execute a block of statements.
- For example: Suppose we want to print “Hello World” 10 times.
- A loop is used for executing a block of statements repeatedly until a particular condition is satisfied.

Looping- for statement

- **Syntax:**

```
for ( initialization; loop Continuation Test; increment/decrement )
    {
    statements;
  }
```

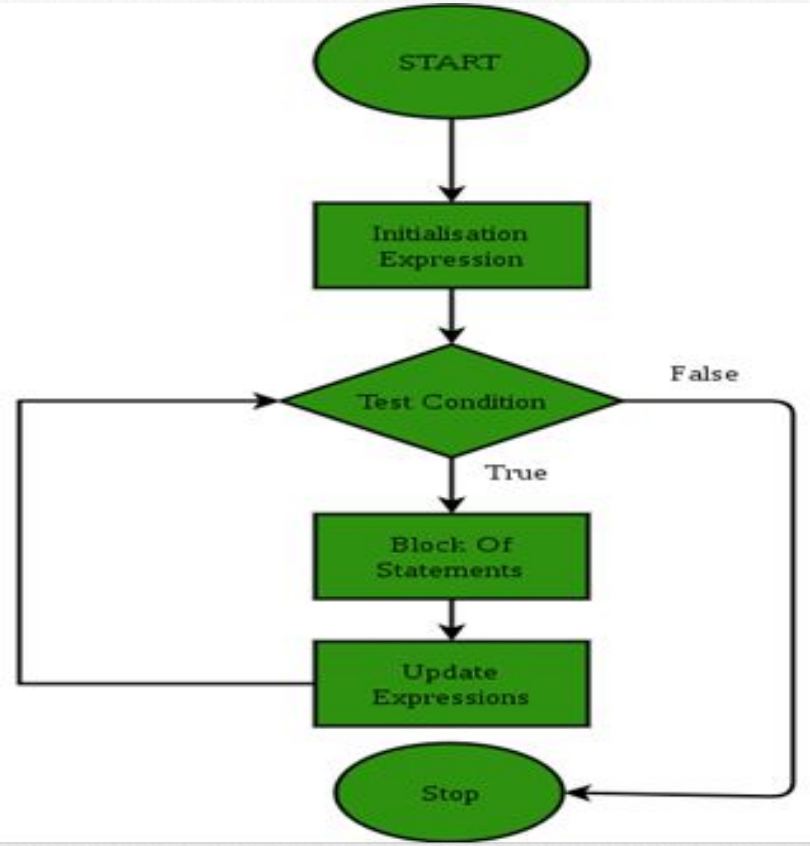
Looping- for statement



Points to remember:

- The initialization step occurs one time only, before the loop begins.
- The condition is tested at the beginning of each iteration of the loop.
  - If the condition is true (non-zero), then the body of the loop is executed next.
  - If the condition is false (zero), then the body is not executed, and execution continues with the code following the loop.
- The incrementation happens **AFTER** the execution of the body, and only when the body is executed.

Flowchart


START
Initialisation Expression
Test Condition
False
True
Block Of Statements
Update Expressions
Stop

Programs

1. Print numbers from 1 to 10
2. Calculate the sum of first n natural numbers.
3. Find factorial of a given number.
4. Printing Multiplication table
5. Find the given number is amstrong number or not.
6. Print the factors of a given number.

Programs

1. Print numbers from 1 to 10

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

Programs

Calculate the sum of first n natural numbers.

```c
#include <stdio.h>
int main() {
    int n, i, sum = 0;

    printf("Enter a positive integer: ");
    scanf("%d", &n);

    for (i = 1; i <= n; ++i) {
        sum += i;
    }

    printf("Sum = %d", sum);
    return 0;
}
```

Programs

Find factorial of a given number.

```c
#include<stdio.h>
int main()
{
    int n;
    int fact=1;
    scanf("%d",&n);

    for(int i=n;i>=2;i--)
    {
        fact=fact*i;
    }
    printf("%d",fact);
}
```

Infinite Loop

```c
#include <stdio.h>

int main () {

   for( ; ; ) {
      printf("This loop will run forever.\n");
   }

   return 0;
}
```

Comma operator

```
#include<stdio.h>
int main()
{
 int i, j = 10, sum;
 for( i = 0, sum = 0; i < 5,j>5; i++, j-- )
     sum += i * j;
printf("%d\n",sum);
}
```

Empty loops

```
#include<stdio.h>
int main()
{
 int i, j = 10, sum;
 for( i = 0, sum = 0; i < 5,j>5; i++, j-- );
    sum += i * j;
printf("%d\n",sum);
}
```

Floating points in loops

```
#include<stdio.h>
int main()
{
 //int i, j = 10, sum;
 for(float x=0.1;x<=2.0;x=x+0.1)
  printf("%f\n",x);
}
```

Floating points counter loops-Infinite loop

```
#include<stdio.h>
int main()
{
 //int i, j = 10, sum;
 for(int x=0;x<=2;x=x+0.1)
  printf("%f\n",x);
}
```

You are climbing a stair case. It takes n steps to reach to the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Example 1:

Input:

2

Output: 2

Explanation: There are two ways to climb to the top. 1. 1 step + 1 step 2. 2 steps

Example 2:

Input:

3

Output:

3

Explanation: There are three ways to climb to the top.

1. 1 step + 1 step + 1 step
2. 1 step + 2 steps
3. 2 steps + 1 step

```
#include <stdio.h>
int main()
{
int a = 1, int b = 2;
int c,n;
scanf("%d", &n);
if( n==1)
      c=1;
else if (n==2)
      c=2;
else
{
 for(int i=3;i<=n;i++)
{
 c = a + b;
 a=b;
b=c;
}
} printf("%d",c);
return 0;}
```

Find the Factor

Determine all positive integer values that evenly divide into a number, its factors. Return the p^{th} element of your list, sorted ascending. If there is no p^{th} element, return *0*.

For example, given the number $n = 20$, its factors are $\{1,2,4,5,10,20\}$. Using **1-based indexing** if $p = 3$, return *4*. If $p > 6$, return *0*.

Sample Input 0

```
10
3
```

Sample Output 0

```
5
```

Explanation 0

Factoring $n = 10$ we get $\{1, 2, 5, 10\}$. We then return the $p = 3^{rd}$ factor as our answer.

Sample Input 1

```
10
5
```

Sample Output 1

```
0
```

Explanation 1

Factoring $n = 10$ we get $\{1, 2, 5, 10\}$. There are only *4* factors and $p = 5$. We return *0* as our answer

```c
#include <stdio.h>
int main() {
	long long int n, p, f = 0, i;
	scanf("%lld", &n);
	scanf("%lld", &p);
	for (i = 1; i <= n; i++)
	{
		if (n % i == 0) {
			f++;
			if (f == p)
				printf("%lld", i);
		}
	}
	if (p > f)
		printf("0");
	return 0;
}
```

Nutrition Value

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For example, food item with value 1 has 1 macronutrient, food item with value 2 has 2 macronutrients, and incrementing in this fashion.

The nutritionist has to recommend the best combination to patients, i.e. maximum total of macronutrients. However, the nutritionist must avoid prescribing a particular sum of macronutrients (an 'unhealthy' number), and this sum is known. The nutritionist chooses food items in the increasing order of their value. Compute the highest total of macronutrients that can be prescribed to a patient, without the sum matching the given 'unhealthy' number.

Here's an illustration:

Given *4* food items (hence value: *1,2,3* and *4*), and the unhealthy sum being *6* macronutrients, on choosing items *1, 2, 3* -> the sum is *6,* which matches the 'unhealthy' sum. Hence, one of the three needs to be skipped. Thus, the best combination is from among:

- *2 + 3 + 4 = 9*
- *1 + 3 + 4 = 8*
- *1 + 2 + 4 = 7*

Since *2 + 3 + 4 = 9,* allows for maximum number of macronutrients, 9 is the right answer.

Complete the code in the editor below. It must return an integer that represents the maximum total of macronutrients, modulo *1000000007* ($10^9 + 7$).

It has the following:

n: an integer that denotes the number of food items
k: an integer that denotes the unhealthy number

```c
#include <stdio.h>
int main() {
	long long int n, k, i, nut = 0;
	scanf("%lld %lld", &n, &k);
	for (i = 1; i <= n; i++)
	{
		nut = nut + i;
		if (nut == k)
			nut = nut - 1;
	}
	printf("%lld", nut % 1000000007);
	return 0;
}
```

Given a number with maximum of 100 digits as input, find the difference between the sum of odd and even position digits.

Input Format:

Take a number in the form of String from stdin.

Output Format:

Print the difference between sum of even and odd digits

Example input:

1453

Output:

1

Explanation:

Here, sum of even digits is 4 + 3 = 7
sum of odd digits is 1 + 5 = 6.
Difference is 1.

Note that we are always taking absolute difference.

Missing Integer Problem

You are given a sequence of n-1 distinct positive integers, all of which are less than or equal to a integer 'n'. You have to find the integer that is missing from the range [1,2, . . . n]. Solve the question without using arrays.

Input Format:

One line containing the integer 'n' where 2<=n<=10,000

First line is followed by a sequence of 'n-1' distinct positive integers. Note that the sequence may not be in any particular order.

Output Format:

One line containing the missing number

Sample Test Cases

Test Case 1

Input

3
1 2

Output

3

MCQs

What will be the output of the C program?

```
#include <stdio.h>
int main()
{
      int i;
      for (i = 0; i > 9; i = i + 3)
      {
            printf("for ");
      }
      return 0;
}
```

A. Nothing prints
B. for
C. for for for
D. None of the above

ANSWER: A

How many times the following for loop will be executed?

```
#include <stdio.h>
int main()
{
      int I = 1;
      for ( ; I <=11 ; )
      {
            I++;
      }
      return 0;
}
```

A. 11 times
B. 10 times
C. 12 times
D. Will Flash Error

ANSWER: A

What will be the final value of i after the execution of the program below?

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

A. 13
B. 12
C. 11
D. 10

ANSWER: A

The value of j will reach to 8 and i will be 13. In the next iteration, j will become 8+13=21 and the condition inside for loop will be invalid, thus the compilation will come out of the loop. So, the value of i will be 13.

What will be the output of the C program?

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

A. Compilation error
B. -128
C. 0
D. 1

ANSWER: B

What will be the output of the C program?

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

A. 127
B. 128
C. Program never ends
D. 0

ANSWER: C

What will be the output of the C program?

```
#include <stdio.h>
int main()
{
      for (5; 2; 2)
            printf("Hello");
      return 0;
}
```

A. Compilation error
B. Program never ends
C. Hello
D. None of the mentioned

ANSWER: B

What will be the output of the C program?

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

A. Compilation error
B. 10
C. Program never ends
D. None of the mentioned

ANSWER: C

What will be the output of the C program?

```
#include <stdio.h>
int main()
{
      int i;
      for (i = 0; i < 0, 5; i++)
            printf("%d ", i);
      return 0;
}
```

A. 1 3
B. Program never ends
C. 1 3 5
D. None of the mentioned

ANSWER: B

What will be the output of given program?

```
#include <stdio.h>
int main()
{
        int a = 3;
        for(;a;printf("%d ", a--));
        return 0;
}
```

A. no output
B. 3 2 1 0
C. 3 2 1
D. infinity loop

ANSWER: C