While loop

Initialization

While(cond)

{----

Inc/dec

}

for Loop

The syntax of the for loop is:

```
for (initializationStatement; testExpression; updateStatement)
{
   // statements inside the body of loop
}
```

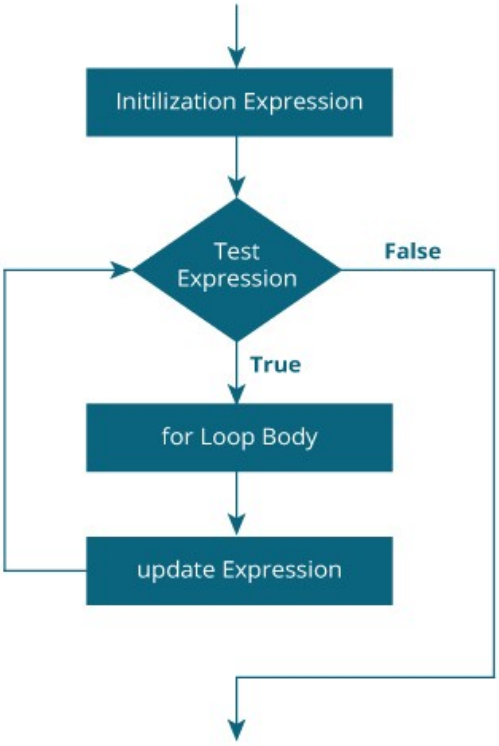
How for loop works?

- The initialization statement is executed only once.
- Then, the test expression is evaluated. If the test expression is evaluated to false, the for loop is terminated.
- However, if the test expression is evaluated to true, statements inside the body of for loop are executed, and the update expression is updated.
- Again the test expression is evaluated.

This process goes on until the test expression is false. When the test expression is false, the loop terminates.

To learn more about test expression (when the test expression is evaluated to true and false), check out relational and logical operators.

for loop Flowchart



Example 1: for loop

```
// Print numbers from 1 to 10
#include <stdio.h>

int main()
 {
  int i;

  for (i = 1; i < 11; ++i)
  {
    printf("%d ", i);
  }
  return 0;
}
```

Output

```
1 2 3 4 5 6 7 8 9 10
```

1. i is initialized to 1.

2. The test expression `i < 11` is evaluated. Since 1 less than 11 is true, the body of `for` loop is executed. This will print the **1** (value of `i`) on the screen.
3. The update statement `++i` is executed. Now, the value of `i` will be 2. Again, the test expression is evaluated to true, and the body of for loop is executed. This will print **2** (value of `i`) on the screen.
4. Again, the update statement `++i` is executed and the test expression `i < 11` is evaluated. This process goes on until `i` becomes 11.
5. When `i` becomes 11, `i < 11` will be false, and the `for` loop terminates.

Example 2: for loop

```
// Program to calculate the sum of first n natural numbers
// Positive integers 1,2,3...n are known as natural numbers

#include <stdio.h>
int main()
{
   int num, count, sum = 0;

   printf("Enter a positive integer: ");
   scanf("%d", &num);   //10

   // for loop terminates w10hen num is less than count
   for(count = 1; count <= num; ++count) // i<=n
   {
      sum =sum + count; // 0+1=1+2=3+3=6+4=10+5=15=55
   }

   printf("Sum = %d", sum); //55
```

```
    return 0;
}
```

Output

```
Enter a positive integer: 10
Sum = 55
```

The value entered by the user is stored in the variable num. Suppose, the user entered 10.

The count is initialized to 1 and the test expression is evaluated. Since the test expression count<=num (1 less than or equal to 10) is true, the body of for loop is executed and the value of sum will equal to 1.

Then, the update statement ++count is executed and the count will equal to 2. Again, the test expression is evaluated. Since 2 is also less than 10, the test expression is evaluated to true and the body of for loop is executed. Now, the sum will equal 3.

This process goes on and the sum is calculated until the count reaches 11.

When the count is 11, the test expression is evaluated to 0 (false), and the loop terminates.

Then, the value of sum is printed on the screen.

Eg. Program to find sum of numbers in the given range.

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

    printf("Enter the numbers in the range: ");
    scanf("%d %d", &m,&n);   //10// 21-25//42-49// m
n


    // for loop termin21ate25 w10hen num is less
than count
    for(count = m; count <= n; ++count) // i<=n
    {
        sum =sum + count; //
0+1=1+2=3+3=6+4=10+5=15=55
```

```
    }

    printf("Sum = %d", sum); //55

    return 0;
}
```

Various forms of for loop in C

I am using variable num as the counter in all the following examples –

1) Here instead of num++, I'm using num=num+1 which is same as num++.

```
for (num=10; num<20; num=num+1)//i=i+2//i++ ++
```

2) Initialization part can be skipped from loop as shown below, the counter variable is declared before the loop.

```
int num=10;
for (;num<20;num++)
```

Note: Even though we can skip initialization part but semicolon (;) before condition is must, without which you will get compilation error.

3) Like initialization, you can also skip the increment part as we did below. In this case semicolon (;) is must after condition logic. In this case the increment or decrement part is done inside the loop.

```
for (num=10; num<20; )
{
    //Statements
```

```
    num++;
}
```

4) This is also possible. The counter variable is initialized before the loop and incremented inside the loop.

```
int num=10;
for (;num<20;)
{
    //Statements
    num++;
}
```

5) As mentioned above, the counter variable can be decremented as well. In the below example the variable gets decremented each time the loop runs until the condition num>10 returns false.

```
for(num=20; num>10; num--)
```

Eg. Program to find sum of even numbers in the given range ie.e 1-20.

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

   printf("Enter the numbers in the range: ");
   scanf("%d ", &n);   //10

   // for loop termin21ate25 w10hen num is less than count
   for(count = 0; count <= n; count=count+2) // i<=n
   {
```

```
      sum =sum + count; // 0+0=0
+2=2+4=6+6=12+8=20+10=30
   }

   printf("Sum = %d", sum); //30

   return 0;
}
```

Nested For Loop in C

We can also have nested `for` loops, i.e one `for` loop inside another `for` loop. Basic syntax is,

```
for(initialization; condition; increment/decrement)
{
   for(initialization; condition;
increment/decrement)
   {
      statement ;
   }
}
```

Lets take an example to understand this:

```
#include <stdio.h>
int main()
{
  for (int i=0; i<2; i++)
  {
```

```
      for (int j=0; j<4; j++)
      {
        printf("%d, %d\n",i ,j);
      }
  }
  return 0;
}
```

Output:

```
0, 0
0, 1
0, 2
0, 3
1, 0
1, 1
1, 2
1, 3
```

In the above example we have a for loop inside another for loop, this is called nesting of loops. One of the example where we use nested for loop is Two dimensional array.

Consider another following example that uses nested for loop in C programming to output a multiplication table:

```
#include <stdio.h>
int main() {
int i, j;
int table = 2;
int max = 5;
for (i = 1; i <= table; i++) { // outer loop
 for (j = 0; j <= max; j++) { // inner loop
   printf("%d x %d = %d\n", i, j, i*j);
 }
 printf("\n"); /* blank line between tables */
}}
```

Output:

```
1 x 0 = 0
```

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5

2 x 0 = 0
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
```

The nesting of for loops can be done up-to any level. The nested loops should be adequately indented to make code readable. In some versions of 'C,' the nesting is limited up to 15 loops, but some provide more.

Example: Program to print half Pyramid of numbers

```c
#include<stdio.h>


void main( )
{
    int i, j;
    /* first for loop */
    for(i = 1; i < 5; i++)
    {
        printf("\n");
        /* second for loop inside the first */
        for(j = i; j > 0; j--)
        {
            printf("%d", j);
        }
    }
}
```

```
1
21
321
4321
```

```
54321
```

Multiple initialization inside for Loop in C

We can have multiple initialization in the for loop as shown below.

```
for (i=1,j=1;i<10 && j<10; i++, j++)
```

What's the difference between above for loop and a simple for loop?

1. It is initializing two variables. Note: both are separated by comma (,).

2. It has two test conditions joined together using AND (&&) logical operator. Note: You cannot use multiple test conditions separated by comma, you must use logical operator such as && or || to join conditions.

3. It has two variables in increment part. **Note:** Should be separated by comma.

Example of for loop with multiple test conditions

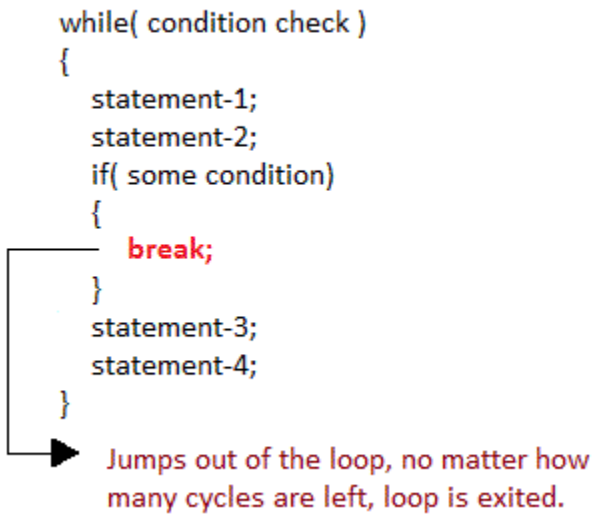
```
#include <stdio.h>
int main()
{
  int i,j;
  for (i=1,j=1 ; i<3 || j<5; i++,j++)
  {
          printf("%d, %d\n",i ,j);
  }
  return 0;
}
```

Jumping Out of Loops

Sometimes, while executing a loop, it becomes necessary to skip a part of the loop or to leave the loop as soon as certain condition becomes **true**. This is known as jumping out of loop.

1) break statement

When break statement is encountered inside a loop, the loop is immediately exited and the program continues with the statement immediately following the loop.



Break Statement in C

The break statement is used mainly in in the switch statement. It is also useful for immediately stopping a loop.

We consider the following program which introduces a break to exit a while loop:

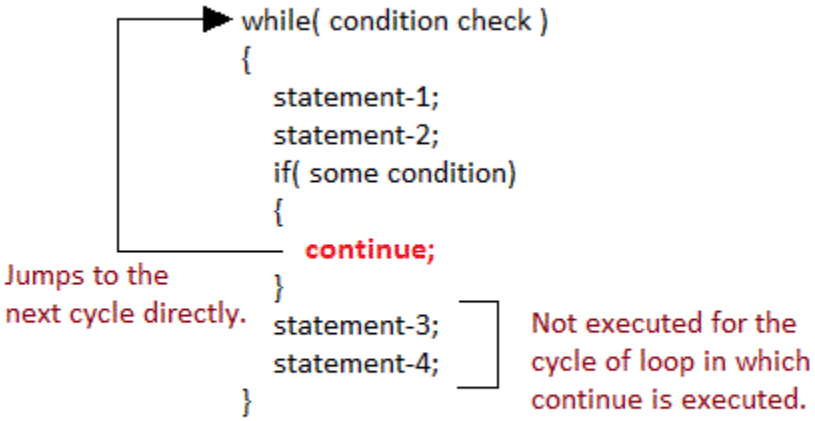
```
#include <stdio.h>
int main() {
int num = 5;
while (num > 0) {
  if (num == 3)
    break;
  printf("%d\n", num);
  num--;
}}
```

Output:

```
5
4
```

2) **Continue Statement in C**

It causes the control to go directly to the test-condition and then continue the loop process. On encountering continue, cursor leave the current cycle of loop, and starts with the next cycle.



When you want to skip to the next iteration but remain in the loop, you should use the continue statement.

For example:

```
#include <stdio.h>
int main() {
int nb = 7;
while (nb > 0) {
  nb--;
  if (nb == 5)
    continue;
 printf("%d\n", nb);
}}
```

Output:

```
6
4
3
2
1
```

So, the value 5 is skipped.

1. Write a program in C to display the first 10 natural numbers. Go to the editor
Expected Output :
1 2 3 4 5 6 7 8 9 10
Click me to see the solution

2. Write a C program to find the sum of first 10 natural numbers. Go to the editor
Expected Output :
The first 10 natural number is :
1 2 3 4 5 6 7 8 9 10

The Sum is : 55
Click me to see the solution

3. Write a program in C to display n terms of natural number and their sum.Go to the editor
Test Data : 7
Expected Output :
The first 7 natural number is :
1 2 3 4 5 6 7
The Sum of Natural Number upto 7 terms : 28
Click me to see the solution

4. Write a program in C to read 10 numbers from keyboard and find their sum and average. Go to the editor
Test Data :
Input the 10 numbers :
Number-1 :2
...
Number-10 :2
Expected Output :
The sum of 10 no is : 55
The Average is : 5.500000
Click me to see the solution

5. Write a program in C to display the cube of the number upto given an integer. Go to the editor
Test Data :
Input number of terms : 5
Expected Output :
Number is : 1 and cube of the 1 is :1
Number is : 2 and cube of the 2 is :8
Number is : 3 and cube of the 3 is :27
Number is : 4 and cube of the 4 is :64
Number is : 5 and cube of the 5 is :125
Click me to see the solution

6. Write a program in C to display the multiplication table of a given integer. Go to the editor
Test Data :
Input the number (Table to be calculated) : 15
Expected Output :
15 X 1 = 15
...
...
15 X 10 = 150
Click me to see the solution

7. Write a program in C to display the multipliaction table vertically from 1 to n. Go to the editor
Test Data :
Input upto the table number starting from 1 : 8
Expected Output :
Multiplication table from 1 to 8

1x1 = 1, 2x1 = 2, 3x1 = 3, 4x1 = 4, 5x1 = 5, 6x1 = 6, 7x1 = 7, 8x1 = 8
...
1x10 = 10, 2x10 = 20, 3x10 = 30, 4x10 = 40, 5x10 = 50, 6x10 = 60, 7x10 = 70, 8x10 = 80
Click me to see the solution

8. Write a program in C to display the n terms of odd natural number and their sum . Go to the editor
Test Data
Input number of terms : 10
Expected Output :
The odd numbers are :1 3 5 7 9 11 13 15 17 19
The Sum of odd Natural Number upto 10 terms : 100
Click me to see the solution

9. Write a program in C to display the pattern like right angle triangle using an asterisk. Go to the editor

The pattern like :

*
**
***
****
Click me to see the solution

10. Write a program in C to display the pattern like right angle triangle with a number. Go to the editor

The pattern like :

1
12
123
1234
Click me to see the solution

11. Write a program in C to make such a pattern like right angle triangle with a number which will repeat a number in a row. Go to the editor

The pattern like :

1
22
333
4444
Click me to see the solution

12. Write a program in C to make such a pattern like right angle triangle with number increased by 1. Go to the editor

The pattern like :

1
2 3
4 5 6

7 8 9 10
Click me to see the solution

13. Write a program in C to make such a pattern like a pyramid with numbers increased by 1. Go to the editor

```
  1
 2 3
4 5 6
7 8 9 10
```

Click me to see the solution

14. Write a program in C to make such a pattern like a pyramid with an asterisk. Go to the editor

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

Click me to see the solution

15. Write a C program to calculate the factorial of a given number. Go to the editor
Test Data :
Input the number : 5
Expected Output :
The Factorial of 5 is: 120
Click me to see the solution

16. Write a program in C to display the n terms of even natural number and their sum. Go to the editor
Test Data :
Input number of terms : 5
Expected Output :
The even numbers are :2 4 6 8 10
The Sum of even Natural Number upto 5 terms : 30
Click me to see the solution

17. Write a program in C to make such a pattern like a pyramid with a number which will repeat the number in the same row. Go to the editor

```
  1
 2 2
3 3 3
4 4 4 4
```

Click me to see the solution

18. Write a program in C to find the sum of the series [1-X^2/2!
+X^4/4!- .....]. Go to the editor
Test Data :
Input the Value of x :2
Input the number of terms : 5
Expected Output :
the sum = -0.415873
Number of terms = 5

value of x = 2.000000
Click me to see the solution

19. Write a program in C to display the n terms of harmonic series and their sum. Go to the editor
1 + 1/2 + 1/3 + 1/4 + 1/5 ... 1/n terms
Test Data :
Input the number of terms : 5
Expected Output :
1/1 + 1/2 + 1/3 + 1/4 + 1/5 +
Sum of Series upto 5 terms : 2.283334
Click me to see the solution

20. Write a program in C to display the pattern like a pyramid using asterisk and each row contain an odd number of asterisks. Go to the editor

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

Click me to see the solution

21. Write a program in C to display the sum of the series [9 + 99 + 999 + 9999 ...]. Go to the editor
Test Data :
Input the number or terms :5
Expected Output :
9 99 999 9999 99999
The sum of the saries = 111105
Click me to see the solution

22. Write a program in C to print the Floyd's Triangle. Go to the editor

1
01
101
0101
10101
Click me to see the solution

23. Write a program in C to display the sum of the series [1+x+x^2/2!+x^3/3! +....]. Go to the editor
Test Data :
Input the value of x :3
Input number of terms : 5
Expected Output :
The sum is : 16.375000
Click me to see the solution

24. Write a program in C to find the sum of the series [x - x^3 + x^5 + ...]. Go to the editor
Test Data :
Input the value of x :2
Input number of terms : 5
Expected Output :
The values of the series:

2
-8
32
-128
512
The sum = 410
Click me to see the solution

25. Write a program in C to display the n terms of square natural number and their sum. Go to the editor
1 4 9 16 ... n Terms
Test Data :
Input the number of terms : 5
Expected Output :
The square natural upto 5 terms are :1 4 9 16 25
The Sum of Square Natural Number upto 5 terms = 55
Click me to see the solution

26. Write a program in C to find the sum of the series 1 +11 + 111 + 1111 + .. n terms. Go to the editor
Test Data :
Input the number of terms : 5
Expected Output :
1 + 11 + 111 + 1111 + 11111
The Sum is : 12345
Click me to see the solution

27. Write a c program to check whether a given number is a perfect number or not. Go to the editor
Test Data :
Input the number : 56
Expected Output :
The positive divisor : 1 2 4 7 8 14 28
The sum of the divisor is : 64
So, the number is not perfect.
Click me to see the solution

28. Write a c program to find the perfect numbers within a given number of range. Go to the editor
Test Data :
Input the starting range or number : 1
Input the ending range of number : 50
Expected Output :
The Perfect numbers within the given range : 6 28
Click me to see the solution

29. Write a C program to check whether a given number is an armstrong number or not. Go to the editor
Test Data :
Input a number: 153
Expected Output :
153 is an Armstrong number.
Click me to see the solution

30. Write a C program to find the Armstrong number for a given range of number. Go to the editor
Test Data :
Input starting number of range: 1
Input ending number of range : 1000
Expected Output :
Armstrong numbers in given range are: 1 153 370 371 407
Click me to see the solution

31. Write a program in C to display the pattern like a diamond. Go to the editor

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

Click me to see the solution

32. Write a C program to determine whether a given number is prime or not. Go to the editor
Test Data :
Input a number: 13
Expected Output :
13 is a prime number.
Click me to see the solution

33. Write a C program to display Pascal's triangle. Go to the editor
Test Data :
Input number of rows: 5
Expected Output :

```
    1
   1  1
  1  2  1
 1  3  3  1
1  4  6  4  1
```

Click me to see the solution
34. Write a program in C to find the prime numbers within a range of numbers. Go to the editor
Test Data :
Input starting number of range: 1
Input ending number of range : 50
Expected Output :
The prime number between 1 and 50 are :
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47
Click me to see the solution

35. Write a program in C to display the first n terms of Fibonacci series. Go to the editor

Fibonacci series 0 1 2 3 5 8 13 .....
Test Data :
Input number of terms to display : 10
Expected Output :
Here is the Fibonacci series upto to 10 terms :
0 1 1 2 3 5 8 13 21 34
Click me to see the solution

36. Write a program in C to display the such a pattern for n number of rows using a number which will start with the number 1 and the first and a last number of each row will be 1. Go to the editor

```
  1
 121
12321
```

Click me to see the solution
37. Write a program in C to display the number in reverse order. Go to the editor
Test Data :
Input a number: 12345
Expected Output :
The number in reverse order is : 54321
Click me to see the solution

38. Write a program in C to check whether a number is a palindrome or not. Go to the editor
Test Data :
Input a number: 121
Expected Output :
121 is a palindrome number.
Click me to see the solution

39. Write a program in C to find the number and sum of all integer between 100 and 200 which are divisible by 9. Go to the editor
Expected Output :
Numbers between 100 and 200, divisible by 9 :
108 117 126 135 144 153 162 171 180 189 198
The sum : 1683
Click me to see the solution

40. Write a C Program to display the pattern like pyramid using the alphabet. Go to the editor

```
      A
    A B A
  A B C B A
A B C D C B A
```

Click me to see the solution

41. Write a program in C to convert a decimal number into binary without using an array. Go to the editor
Test Data :
Enter a number to convert : 25
Expected Output :

The Binary of 25 is 11001.
Click me to see the solution

42. Write a program in C to convert a binary number into a decimal number without using array, function and while loop. Go to the editor
Test Data :
Input a binary number :1010101
Expected Output :
The Binary Number : 1010101
The equivalent Decimal Number : 85
Click me to see the solution

43. Write a C program to find HCF (Highest Common Factor) of two numbers. Go to the editor
Test Data :
Input 1st number for HCF: 24
Input 2nd number for HCF: 28
Expected Output :
HCF of 24 and 28 is : 4
Click me to see the solution

44. Write a program in C to find LCM of any two numbers using HCF. Go to the editor
Test Data :
Input 1st number for LCM: 15
Input 2nd number for LCM: 20
Expected Output :
The LCM of 15 and 20 is : 60
Click me to see the solution

45. Write a program in C to find LCM of any two numbers. Go to the editor
Test Data :
Input 1st number for LCM: 15
Input 2nd number for LCM: 20
Expected Output :
The LCM of 15 and 20 is : 60
Click me to see the solution

46. Write a program in C to convert a binary number into a decimal number using math function. Go to the editor
Test Data :
Input the binary number :1010100
Expected Output :
The Binary Number : 1010100
The equivalent Decimal Number is : 84
Click me to see the solution

47. Write a C program to check whether a number is a Strong Number or not. Go to the editor
Test Data :
Input a number to check whether it is Strong number: 15
Expected Output :

15 is not a Strong number.
Click me to see the solution

48. Write a C program to find Strong Numbers within a range of numbers. Go to the editor
Test Data :
Input starting range of number : 1
Input ending range of number: 200
Expected Output :
The Strong numbers are :
1 2 145
Click me to see the solution

49. Write a c program to find out the sum of an A.P. series. Go to the editor
Test Data :
Input the starting number of the A.P. series: 1
Input the number of items for the A.P. series: 10
Input the common difference of A.P. series: 4
Expected Output :
The Sum of the A.P. series are :
1 + 5 + 9 + 13 + 17 + 21 + 25 + 29 + 33 + 37 = 190
Click me to see the solution

50. Write a program in C to convert a decimal number into octal without using an array. Go to the editor
Test Data :
Enter a number to convert : 79
Expected Output :
The Octal of 79 is 117.
Click me to see the solution

51. Write a program in C to convert an octal number to a decimal without using an array. Go to the editor
Test Data :
Input an octal number (using digit 0 - 7) :745
Expected Output :
The Octal Number : 745
The equivalent Decimal Number : 485
Click me to see the solution

52. Write a program in c to find the Sum of GP series. Go to the editor
Test Data :
Input the first number of the G.P. series: 3
Input the number or terms in the G.P. series: 5
Input the common ratio of G.P. series: 2
Expected Output :
The numbers for the G.P. series:
3.000000 6.000000 12.000000 24.000000 48.000000
The Sum of the G.P. series : 93.000000
Click me to see the solution

53. Write a program in C to convert a binary number to octal. Go to the editor
Test Data :

Input a binary number :1001
Expected Output :
The Binary Number : 1001
The equivalent Octal Number : 11
Click me to see the solution

54. Write a program in C to convert an octal number into binary. Go to the editor
Test Data :
Input an octal number (using digit 0 - 7) :57
Expected Output :
The Octal Number : 57
The equivalent Binary Number : 101111

Click me to see the solution

55. Write a program in C to convert a decimal number to hexadecimal. Go to the editor
Test Data :
Input any Decimal number: 79
Expected Output :
The equivalent Hexadecimal Number : 4F
Click me to see the solution

56. Write a program in C to Check Whether a Number can be Express as Sum of Two Prime Numbers. Go to the editor
Test Data :
Input a positive integer: 16
Expected Output :
16 = 3 + 13
16 = 5 + 11
Click me to see the solution

57. Write a program in C to print a string in reverse order. Go to the editor
Test Data :
Input a string to reverse : Welcome
Expected Output :
Reversed string is: emocleW
Click me to see the solution

58. Write a C program to find the length of a string without using the library function. Go to the editor
Test Data :
Input a string : welcome
Expected Output :
The string contains 7 number of characters.
So, the length of the string welcome is : 7
Click me to see the solution

59. Write a program in C to check Armstrong number of n digits. Go to the editor
Test Data :
Input an integer : 1634
Expected Output :
1634 is an Armstrong number
Click me to see the solution

https://www.w3resource.com/c-programming-exercises/for-loop/index.php