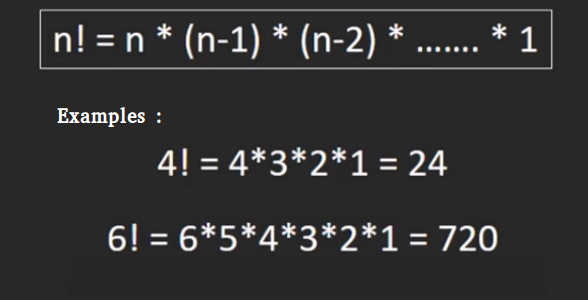
Program for factorial of a number

Factorial of a non-negative integer, is multiplication of all integers smaller than or equal to n. For example factorial of 6 is 6\*5\*4\*3\*2\*1 which is 720.



**Recursive Solution:**  
Factorial can be calculated using following recursive formula.

n! = n \* (n-1)!

n! = 1 if n = 0 or n = 1

[**Recommended: Please solve it on “*PRACTICE* ” first, before moving on to the solution.**](https://practice.geeksforgeeks.org/problems/factorial/0)

Following is implementation of factorial.

* C
* Java
* Python3
* C#
* PHP

|  |
| --- |
| // C program to find factorial of given number  #include<stdio.h>    // function to find factorial of given number  unsigned int factorial(unsigned int n)  {      if (n == 0)        return 1;      return n\*factorial(n-1);  }    int main()  {      int num = 5;      printf("Factorial of %d is %d", num, factorial(num));      return 0;  } |

Run on IDE

Output:

Factorial of 5 is 120

**Iterative Solution:**  
Factorial can also be calculated iteratively as recursion can be costly for large numbers.

* C
* Java
* Python3
* C#
* PHP

|  |
| --- |
| #include<stdio.h>    // function to find factorial of given number  unsigned int factorial(unsigned int n)  {      int res = 1, i;      for (i=2; i<=n; i++)          res \*= i;      return res;  }    int main()  {      int num = 5;      printf("Factorial of %d is %d", num, factorial(num));      return 0;  } |

Run on IDE

**Output :**

Factorial of 5 is 120

Time complexity of the above solutions is O(n).

**One line Solution (Using Ternary operator):**

* C++
* Java
* Python3
* C#
* PHP

|  |
| --- |
| // C++ program to find factorial of given number  #include<iostream>    int factorial(int n)  {      // single line to find factorial      return (n==1 || n==0) ? 1: n \* factorial(n - 1);  }    // Driver Code  int main()  {      int num = 5;      printf ("Factorial of %d is %d", num, factorial(num));      return 0;  }    // This code is contributed by  Rithika palaniswamy. |

Run on IDE

Output:

Factorial of 5 is 120

# Factorial of a large number

**How to compute factorial of 100 using a C/C++ program?**  
Factorial of 100 has 158 digits. It is not possible to store these many digits even if we use long long int.

**Examples :**

Input : 100

Output : 933262154439441526816992388562667004-

907159682643816214685929638952175999-

932299156089414639761565182862536979-

208272237582511852109168640000000000-

00000000000000

Input :50

Output : 3041409320171337804361260816606476884-

4377641568960512000000000000

Following is a simple solution where we use an array to store individual digits of the result. The idea is to use basic mathematics for multiplication.

The following is detailed algorithm for finding factorial.

***factorial(n)***  
1) Create an array ‘res[]’ of MAX size where MAX is number of maximum digits in output.  
2) Initialize value stored in ‘res[]’ as 1 and initialize ‘res\_size’ (size of ‘res[]’) as 1.  
3) Do following for all numbers from x = 2 to n.  
……a) Multiply x with res[] and update res[] and res\_size to store the multiplication result.

***How to multiply a number ‘x’ with the number stored in res[]?***  
The idea is to use simple school mathematics. We one by one multiply x with every digit of res[]. The important point to note here is digits are multiplied from rightmost digit to leftmost digit. If we store digits in same order in res[], then it becomes difficult to update res[] without extra space. That is why res[] is maintained in reverse way, i.e., digits from right to left are stored.

***multiply(res[], x)***  
1) Initialize carry as 0.  
2) Do following for i = 0 to res\_size – 1  
….a) Find value of res[i] \* x + carry. Let this value be prod.  
….b) Update res[i] by storing last digit of prod in it.  
….c) Update carry by storing remaining digits in carry.  
3) Put all digits of carry in res[] and increase res\_size by number of digits in carry.

**Example to show working of multiply(res[], x)**

A number 5189 is stored in res[] as following.

res[] = {9, 8, 1, 5}

x = 10

Initialize carry = 0;

i = 0, prod = res[0]\*x + carry = 9\*10 + 0 = 90.

res[0] = 0, carry = 9

i = 1, prod = res[1]\*x + carry = 8\*10 + 9 = 89

res[1] = 9, carry = 8

i = 2, prod = res[2]\*x + carry = 1\*10 + 8 = 18

res[2] = 8, carry = 1

i = 3, prod = res[3]\*x + carry = 5\*10 + 1 = 51

res[3] = 1, carry = 5

res[4] = carry = 5

res[] = {0, 9, 8, 1, 5}

Below is the implementation of above algorithm.

* C++
* Java
* Python
* C#

|  |
| --- |
| // C++ program to compute factorial of big numbers  #include<iostream>  using namespace std;    // Maximum number of digits in output  #define MAX 500    int multiply(int x, int res[], int res\_size);    // This function finds factorial of large numbers  // and prints them  void factorial(int n)  {      int res[MAX];        // Initialize result      res[0] = 1;      int res\_size = 1;        // Apply simple factorial formula n! = 1 \* 2 \* 3 \* 4...\*n      for (int x=2; x<=n; x++)          res\_size = multiply(x, res, res\_size);        cout << "Factorial of given number is \n";      for (int i=res\_size-1; i>=0; i--)          cout << res[i];  }    // This function multiplies x with the number  // represented by res[].  // res\_size is size of res[] or number of digits in the  // number represented by res[]. This function uses simple  // school mathematics for multiplication.  // This function may value of res\_size and returns the  // new value of res\_size  int multiply(int x, int res[], int res\_size)  {      int carry = 0;  // Initialize carry        // One by one multiply n with individual digits of res[]      for (int i=0; i<res\_size; i++)      {          int prod = res[i] \* x + carry;            // Store last digit of 'prod' in res[]          res[i] = prod % 10;            // Put rest in carry          carry  = prod/10;      }        // Put carry in res and increase result size      while (carry)      {          res[res\_size] = carry%10;          carry = carry/10;          res\_size++;      }      return res\_size;  }    // Driver program  int main()  {      factorial(100);      return 0;  } |

Run on IDE

**Output :**

Factorial of given number is

9332621544394415268169923885626670049071596826438162146859296389

5217599993229915608941463976156518286253697920827223758251185210

916864000000000000000000000000