

FACTORIAL

$$N = 5$$

$$\text{ans} : \underline{120}$$

$$n! = n \times (n-1) \times (n-2) \times \dots \times 1$$

$$= 1 \times 2 \times 3 \times \dots \times n$$

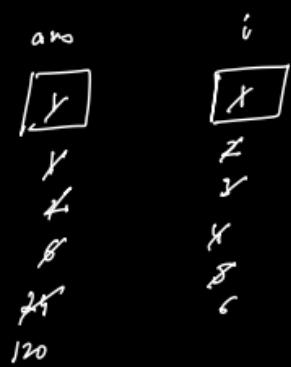
$$3! = 3 \times 2 \times 1 = 6$$

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

```
int N;  
cin >> N;
```

$$\boxed{5}$$

```
int ans = 1;  
for(int i=1 ; i <= N; i++)  
{  
    ans *= i;  
}  
cout << ans;
```



BINOMIAL COEFFICIENT

$${}^n C_r = \frac{n!}{r!(n-r)!}$$

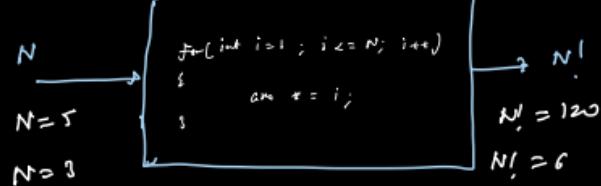
Input
 $n = 5$
 $r = 2$

$${}^5 C_2 = \frac{5!}{2! \times 3!} = \frac{120}{2 \times 6} = 10$$

if p : n and r

o/p : ${}^n C_r = \frac{n!}{r!(n-r)!}$

int N, R;
cin >> N >> R;



int nFact = 1;

for([int i=1; i <= N; i++)
{
 nFact * = i;
}
] N!

```

int rFact = 1;
for( int i=1 ; i <= R ; i++ ) {
    rFact * = i;
}

```

```

int nrFact = 1;
for( int i=1 ; i <= N-R ; i++ ) {
    nrFact * = i;
}

```

cout << nFact / (rFact * nrFact);

FUNCTIONS

- Set of statements
- Write only once
- Reuse it multiple times
- Makes code:
 - Cleaner
 - Shorter
 - Easier to debug



```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     int n;
7     cin >> n;
8
9     int ans = 1;
10    for(int i = 1; i <= n; i++)
11    {
12        ans *= i;
13    }
14
15    cout << ans << endl;
16 }
17 |
```

```
7     cin >> n >> r;
8
9     int nFact = 1;
10    for(int i = 1; i <= n; i++)
11    {
12        nFact *= i;
13    }
14
15    int rFact = 1;
16    for(int i = 1; i <= r; i++)
17    {
18        rFact *= i;
19    }
20
21    int nrFact = 1;
22    for(int i = 1; i <= n - r; i++)
23    {
24        nrFact *= i;
25    }
26
27 }
```

```
2 using namespace std;
3
4 int main()
5 {
6     int n, r;
7     cin >> n >> r;
8
9     // n!
10    int nFact = 1;
11    for(int i = 1; i <= n; i++)
12    {
13        nFact *= i;
14    }
15
16    // r!
17    int rFact = 1;
18    for(int i = 1; i <= r; i++)
19    {
20        rFact *= i;
21    }
22    |
23    int nrFact = 1;
24    for(int i = 1; i <= n - r; i++)
25    {
26        nrFact *= i;
27    }
28
29    cout << nFact / (rFact * nrFact) << endl;
30 }
```



Webinar chat

more practice problems(codeforces group) today for those who are comfortable in todays class

Ganesh Mhetre to Everyone

no

Siddharth Singh Rawat to Everyone

return

Aryan Gupta to Everyone

what if we return nothing

Mrigank Manu to Everyone

ncr

Abhinav Kumar to Everyone

aage se alu piche se sona

Karan Bhatt to Everyone

more practice problems(codeforces group) today for those who are comfortable in todays class

more practice problems(codeforces group) today for those who are comfortable in todays class

Hemant Rathore to Everyone

name change?

Mrigank Manu to Everyone

ncr

Who can see your messages? Recording on

Chat disabled

```
1 #include <iostream>
2 using namespace std;
3
4 int factorial(int n)
5 {
6     int ans = 1;
7     for(int i = 1; i <= n; i++)
8     {
9         ans *= i;
10    }
11    return ans;
12 }
13
14 int main()
15 {
16     int n, r;
17     cin >> n >> r;
18
19     // n!
20     int nFact = factorial(n);
21
22     // r!
23     int rFact = factorial(r);
24
25     // (n-r)!
26     int nrFact = factorial(n - r);
27
28     cout << nFact / (rFact * nrFact) << endl;
29 }
```

FUNCTION SYNTAX

```
return_type function_name(parameters)
{
    // function body
    return value;
}
```

- Return type → what the function gives back
- Function name
- Parameters → inputs
- Return statement

```
int factorial ( int n )
{
    int ans = 1;
    for( i=1; i<=n; i++ )
    {
        ans *= i;
    }
    return ans;
}
```

EXAMPLES

```
int sum2(int a, int b)
{
    return a + b;
}
```

```
int sum3(int a, int b, int c)
{
    return a + b + c;
}
```

```
int factorial(int n)
{
    int ans = 1;
    for( int i = 1; i <= n; i++ )
    {
        ans *= i;
    }
    return ans;
}
```

```
1 #include <iostream>
2 using namespace std;
3
4 int sum2(int a, int b)
5 {
6     return a + b;
7 }
8
9 int main()
10 {
11     int ans1 = sum2(2, 5);
12     int ans2 = sum2(10, 2);
13     int ans3 = sum2(1, 5);
14
15     cout << ans1 << " " << ans2 << " " << ans3 << endl;
16
17 }
```



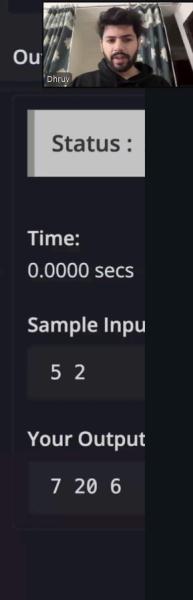
EXAMPLES

```
void print1toN(int n)
{
    for(int i = 1; i <= n; i++)
    {
        cout << i << endl;
    }
}
```

```
void printSquare(int n, char ch)
{
    for(int i = 1; i <= n; i++)
    {
        for(int j = 1; j <= n; j++)
        {
            cout << ch;
        }
        cout << endl;
    }
}
```

x x x x x
x x x x x
x x x x x
x x x x x

```
2 using namespace std;
3
4 // write a function that takes one input integer n
5 // task of this function is to print numbers from 1 to n
6
7 void print1toN(int n)
8 {
9     for(int i = 1; i <= n; i++)
10    {
11        cout << i << " ";
12    }
13    cout << endl;
14 }
15
16
17 int main()
18 {
19
20     cout << ans1 << " " << ans2 << " " << ans3 << endl;
21
22 }
```



```
void printSquare(int n, char ch)
{
    for(int i = 1; i <= n; i++)
    {
        for(int j = 1; j <= n; j++)
        {
            cout << ch;
        }
        cout << endl;
    }
}

int main()
{
    printSquare(5, '*');
}
```

FUNCTION RULES

- Number of Parameters Must Match
- Return Type Must Match
- Return Ends the Function
- A Function May or May Not Return
 - int, double, bool → returns value
 - void → prints only

```
1 #include <iostream>
2 using namespace std;
3
4 bool sum(int a, int b)
5 {
6     return a + b;
7 }
8
9
10 void printSquare(int n, char ch)
11 {
12     for(int i = 1; i <= n; i++)
13     {
14         for(int j = 1; j <= n; j++)
15         {
16             cout << ch;
17         }
18         cout << endl;
19     }
20 }
21
22 int main()
23 {
24     int ans = sum(2, 3);
25
26     cout << ans << endl; // 5
27
28     printSquare(5, '@');
29
30     printSquare(10, '#');
31
32     printSquare(7, '-');
33 }
34
35
```

5 2

Output

Status : Successfully executed

Time: 0.0000 secs Memory: 3.596 Mb

Sample Input

5 2

Your Output

1
@eeee
@eeee
@eeee
@eeee
@eeee

#####

FUNCTION RULES

- Number of Parameters Must Match
 - Return Type Must Match
 - Return Ends the Function *Call*
 - A Function May or May Not Return
 - int, double, bool → returns value
 - void → prints only

Online C++ Compiler and Debugger

C++ Run Visualize Code

```
1 #include <iostream>
2 using namespace std;
3
4 void f(int n)
5 {
6     cout << "Hello, I am function f" << endl;
7     return;
8
9
10    cout << n << endl;
11
12    cout << |
13 }
14
15 int main()
16 {
17     f(10);
18
19     f(20);
20 }
21
22
```

Output

Status : Successfully executed

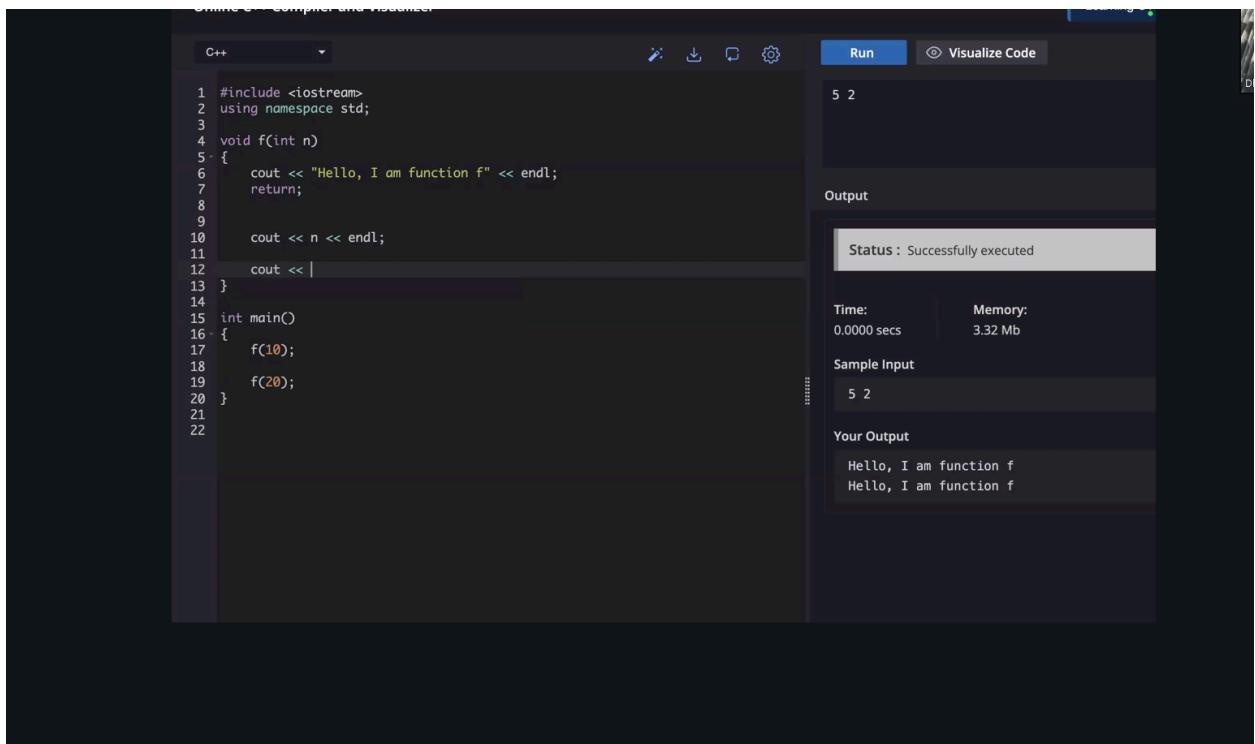
Time: 0.0000 secs Memory: 3.32 Mb

Sample Input

5 2

Your Output

Hello, I am function f
Hello, I am function f



Q: How many factors of a number?

1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6 int n;
7 cin >> n;
8
9 int cnt = 0;
10 for(int i = 1; i <= n; i++)
11 {
12 // i is a factor of n
13 if(n % i == 0)
14 {
15 cnt++;
16 }
17 }
18
19 cout << cnt << endl;
20 }
21
22

Output

Status : Successfully executed

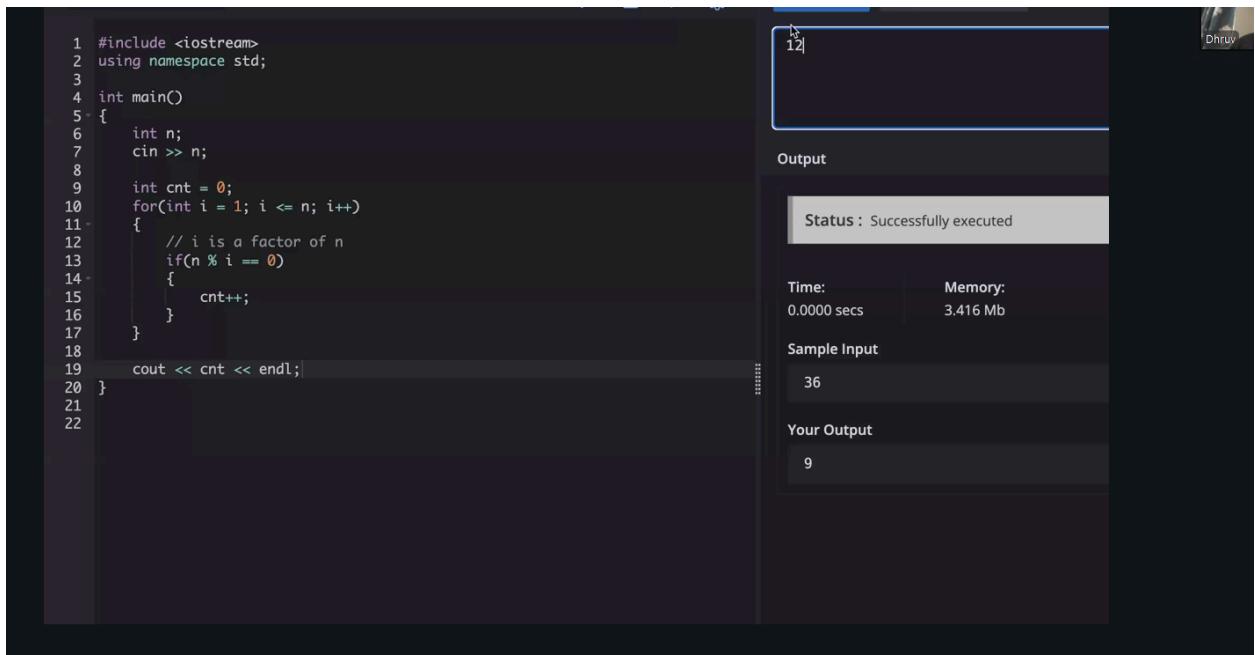
Time: 0.0000 secs Memory: 3.416 Mb

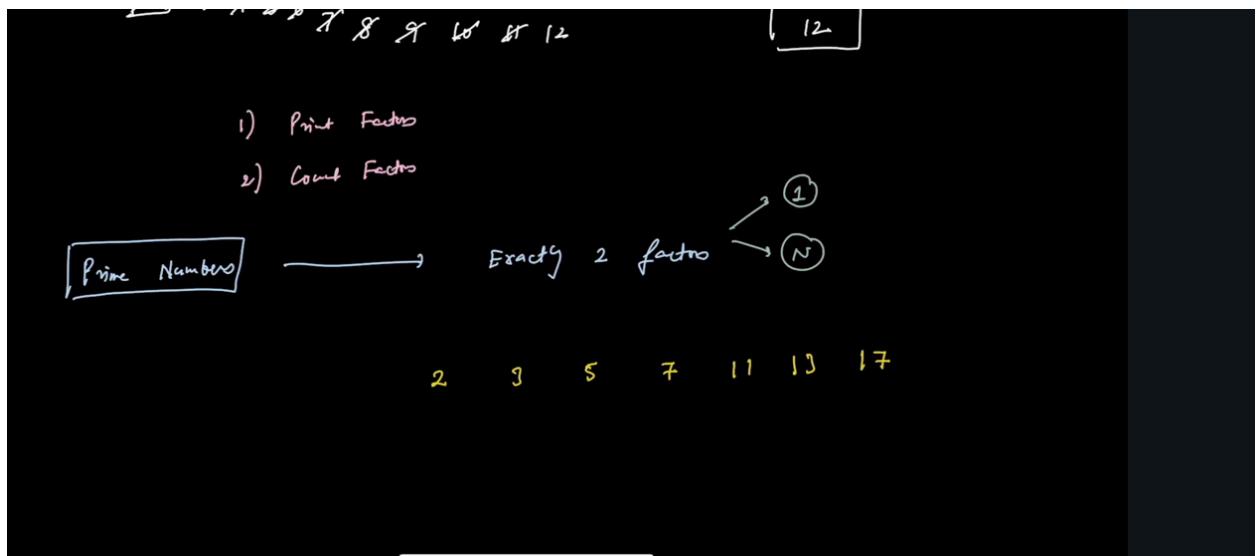
Sample Input

36

Your Output

9





Q: Any particular number is prime or not??

Screenshot of a C++ code editor showing a program to check if a number is prime. The code uses a loop to count factors and then checks if the count is exactly 2.

```
#include <iostream>
using namespace std;

int main()
{
    int n;
    cin >> n;

    int cnt = 0;
    for(int i = 1; i <= n; i++)
    {
        // i is a factor of n
        if(n % i == 0)
        {
            cnt++;
        }
    }

    cout << cnt << endl;

    if(cnt == 2)
    {
        cout << "Prime";
    }
    else
    {
        cout << "Not prime";
    }
}
```

The output window shows the program's output for the input "2", which is "2 Prime".

Status: Successfully executed

Time: 0.0000 secs | Memory: 3.38 Mb

Sample Input: 2

Your Output: 2
Prime

```
4 // true / false
5 bool isPrime(int n)
6 {
7     int cnt = 0;
8     for(int i = 1; i <= n; i++)
9     {
10         // i is a factor of n
11         if(n % i == 0)
12         {
13             cnt++;
14         }
15     }
16     if(cnt == 2)
17     {
18         return true;
19     }
20     else
21     {
22         return false;
23     }
24 }
25 }
26
27
28 int main()
29 {
30     int n;
31     cin >> n;
32
33     bool ans = isPrime(n);
34
35     if(ans)
36     {
37         cout << |
38     }
39 }
40 }
```

Output

Status : Successfully executed

Time: 0.0000 secs | Memory: 3.528 Mb

Sample Input

1

Your Output

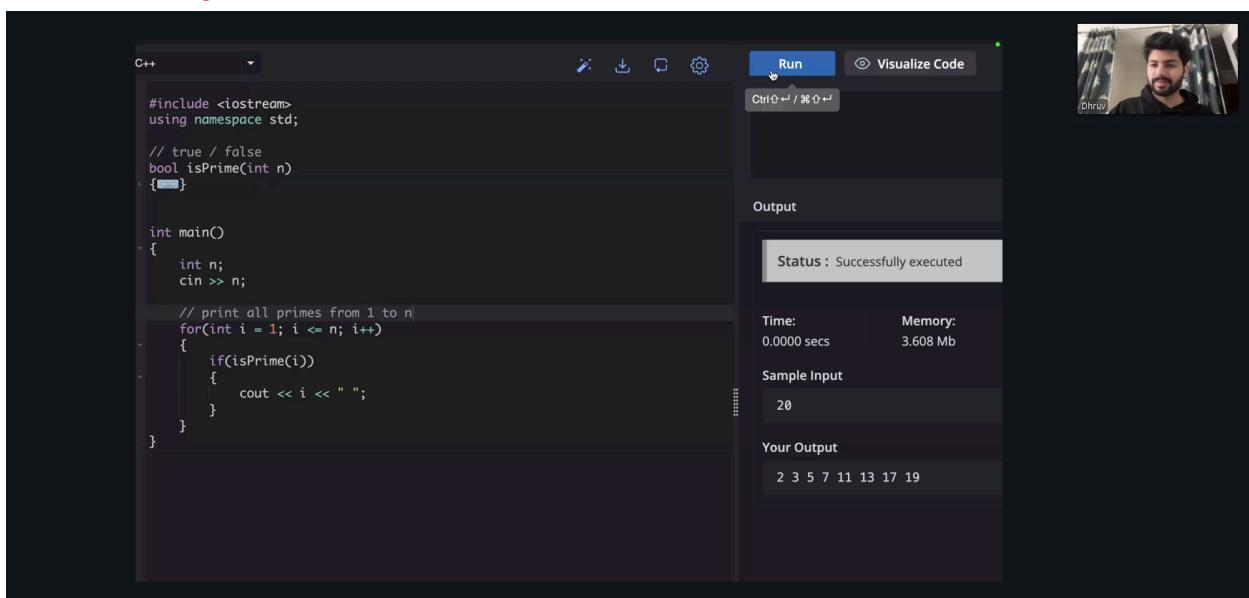
1
Not prime

```

1 // true / false
5 bool isPrime(int n)
6 {
7     int cnt = 0;
8     for(int i = 1; i <= n; i++)
9     {
10         // i is a factor of n
11         if(n % i == 0)
12         {
13             cnt++;
14         }
15     }
16
17     return cnt == 2;
18 }
19
20
21 int main()
22 {
23     int n;
24     cin >> n;
25
26     bool ans = isPrime(n);
27
28     if(ans)
29 {

```

Q: Print all prime number from 1 to N



The screenshot shows a code editor interface with a dark theme. On the left, there is a code editor window containing C++ code. The code defines a boolean function `isPrime` and a `main` function that prints prime numbers from 1 to `n`. In the center, there is a toolbar with a "Run" button highlighted. To the right of the toolbar is an "Output" panel. The output panel displays the following information:

- Status: Successfully executed
- Time: 0.0000 secs
- Memory: 3.608 Mb
- Sample Input: 20
- Your Output: 2 3 5 7 11 13 17 19

A small video thumbnail of a person is visible in the top right corner of the interface.