

Q. Given  $n$  and  $r$ . find the value of  $nCr$

formula  $\rightarrow$

$$nCr \rightarrow \frac{n!}{(n-r)! \cdot r!} \rightarrow \frac{5!}{(5-2)! \cdot 2!} = \frac{120}{2! \cdot 2!} = \frac{120}{4} = 30$$

step to find  $n!$

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

$$\downarrow$$

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

code to find fact  $\rightarrow$

```
int fact = 1;
for (int i = 1; i <= n; i++) {
  fact = fact * i;
}
sop(fact);
```

do not use function.

Left

```
int n = 4, r = 2;
```

```
int fact = 1; // n!  
for (int i = 1; i <= n; i++)  
    fact = fact * i;  
sop(fact);
```

```
int fact = 1; // (n-r)!  
for (int i = 1; i <= (n-r); i++)  
    fact = fact * i;  
sop(fact);
```

```
int fact2 = 1; // r!  
for (int i = 1; i <= r; i++)  
    fact2 = fact2 * i;  
sop(fact);
```

```
int ans = fact / (fact1 * fact2);  
sop(ans);
```

Problems / issues

- ↳ Code repetition
- ↳ Error prone
- ↳ Maintenance is easy
- ↳ lengthy.

Right

```
int fact = getfactorial(n);  
int fact1 = getfactorial(n-r);  
int fact2 = getfactorial(r);  
int ans = fact / (fact1 * fact2);
```

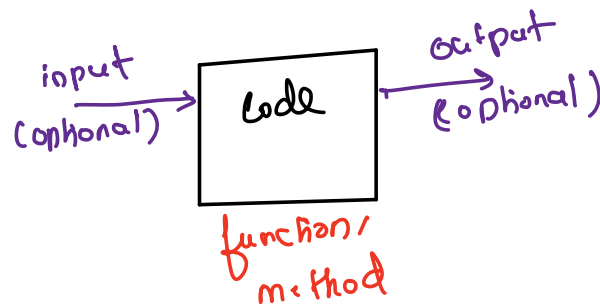
```
getfactorial(n) {  
    int fact = 1;  
    for (int i = 1; i <= n; i++)  
        fact = fact * i;  
    return fact;  
}
```



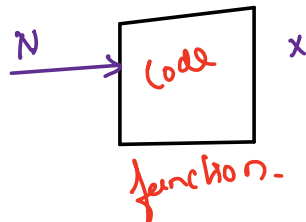
## Syntax of function / method

— predefined  $\leftarrow$  returnType functionName (type input)  $\rightarrow$  given by user

return \_\_\_\_\_ ; (optional);

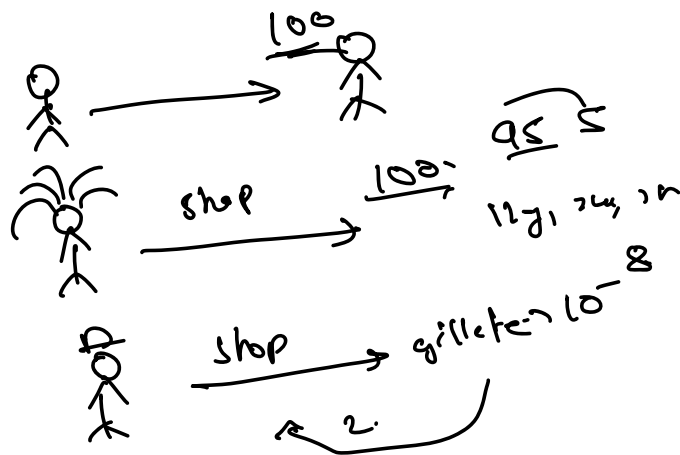


Q.1) Given a number  $N$ . write a function that Prints the factorial of that number



Sol<sup>n</sup>

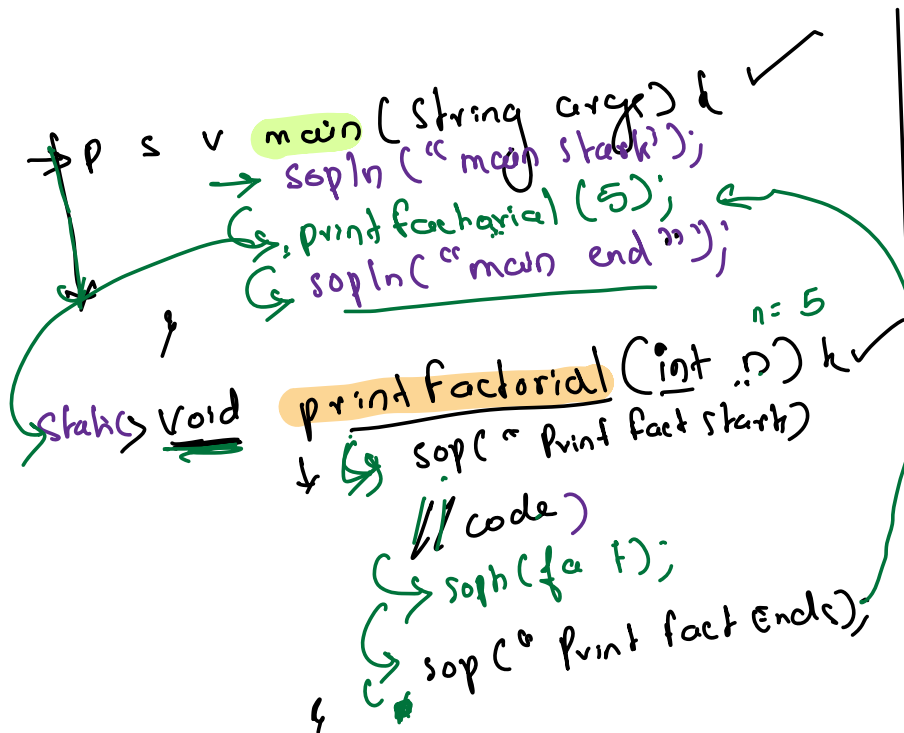
```
- void printfactorial(int n){  
    int fact=1;  
    for(int i=1; i<=n; i++){  
        fact = fact * i;  
    }  
    sop(fact);  
}
```



main (String args) {  
 printfactorial(5);  
}

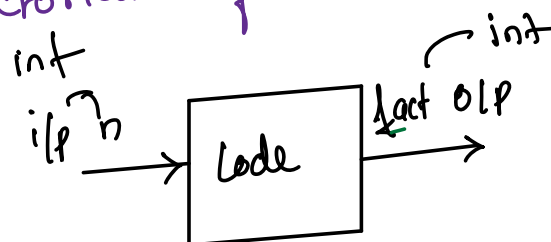
printfactorial(int n) { n=5  
 sop("Print fact start")  
 // code  
 sop(fact);  
 sop("Print fact ends");  
 go back to main

Print fact start  
120  
Print fact ends



Or  
 main starts  
 Print fact starts.  
 120  
 Print fact ends  
 main ends.

Q. Given  $n$ . write a function which will return factorial of that number.



```
static int getFactorial (int n) {
    int fact = 1;
    for (int i = 1; i <= n; i++) {
        fact = fact * i;
    }
    return fact;
}
```

```

p s v main (storage --) {
    get factorial (6);
}

```

Ques 2

```

static int sum (int a, int b) {
    return a + b; // 15.
}

psumc() {
    sum(5, 10)
}

```

Diagram: An arrow from the `sum(5, 10)` call in `psumc()` points to the `return a + b; // 15.` line in the `sum` function. Another arrow from the `return` statement points to the value `15` written below the function call.

a) 15

b) C.E

c) None.

Ques 22

```

static void int sum (int a, int b) {
    return a + b; // 15.
    // remove return
    psumc() {
        sum(5, 10)
    }
}

```

Diagram: A large arrow points from the `static void` declaration to the `int` type annotation above it. Another arrow points from the `return a + b; // 15.` line to the `// remove return` comment below it.

a) 15

b) C.E

c) None.

Ques 23

```

static int sum(int a, int b) {
    return a + b; // 15.
}

psumc() {
    sop(sum(s, 10)); // o/p 15.
}

```

a) 15

b) C.E

c) None.

```

static int sum(int a, int b) {
    return a + b + 10;
    // 5 + 10 + 10 // 25.
}

```

```

psumc() {
    int ans = sum(s, 10);
    sop(ans);
}

```

a) 15

b) C.E

c) None.

d) 25.

$a = -6$   
 static int square(int a) {  
     return  $a * a$ ;  
     //  $+6 * -6 = -36$   
 }  
 printf("square(-6) = %d", square(-6));  
 }

a) 36

b) -36

c) None

d) error



```
static int square(int x) {
    return x * x;
}
```

```
static int sum(int a, int b) {
    return a + b; // 15.
}
```

```
psumc) {
```

15

```
int n1 = square(3); // 9, n1
int n2 = square(5); // 25, n2
```

→ sofn( add(n1, square(9));  
add(9 + 81)

90 o/p

a) 90

```

public static void product    }
    int c = a * b;
    sop(c)
}

```

```

p sum ( ) {
    int c = product(2,5); //
}

```

← void

int is not compatible with  
typ void

Error

```

public static int func(int a) {
    ↪ println("def");
    ↪ return;
    sop(a + "ghi"); // is not reachable.
}

psum { } {
    sop("abc");
    func(10);
}

```

returning

Ans Error

Quiz

```
public static void produe (int a, int b) {  
    int c = a * b;  
    sop c;  
}
```

return c; → this is errors

```
psum (    ) {  
  
}
```

Doubt session

getSum(10, 20);

getSum

```
{  
    int a = 10;  
    int b = 20;
```

```
    int a = 20 erro  
    b = 60;  
}
```

```
public static int getSum() {  
    int a = 10;  
    int b = 20;  
    return a + b; // 30  
}
```

```
public static void getSum(int a, int b) {  
    int a = 10;  
    b = 20;  
    return a + b; // 30  
}
```

problem solving

error

```
public static int getSum() {  
    int a = 10;  
    int b = 20;  
    sop(a+b)  
    int c;  
    int d;  
    for ( →  
        {  
        }  
    }  
}
```

beginner, intermediate, advanced 7-9  
↓  
useful  
Amal  
path, oops  
goal  
aim

num = 1056  
+ 1000  
-----  
2056

sop( )

O/P  
[ ]

sop product (2,8);

```
int product = 2 * 5;
```

$$\text{sup}(C \text{ product});$$

public class Solution {

(public int solkr1(int A, B) {  
return A + B;

7 9

$\log \ln(\text{select } \frac{1}{1})$

for ( )

Boz

pred.

For

for c)

for c)

low



<https://www.interviewbit.com/snippet/67f735df709eb16f98bd/>