

Note:- We can make same name of function but cond'n only is the parameter data type is different

ex:- Swap(int a, int b), Swap(Float a, Float b).

classmate

Date _____

Page _____

Function Overloading:-

DAY 5

* Function with Examples.

function provide Abstraction.

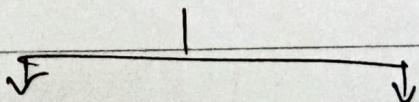
For example:- we can use library functions without worrying about their internal work.

Abstraction:-

Abstraction in programming is the process of hiding the complex implementation details of a particular functionality.

ex:- swap, sqrt ^{in python}

Function



change

User Defined
Function

'f's are used
to reduce
complexity of big
problems.

(usee)

Library
Function.

They are built in Functions.
It's are used directly
without defining because
that already defined & by
consist of compiler package.

* Function challenges by Aman Dhat. 6.2 & 6.3

1) Print all Prime Numbers between 2 given numbers

```
#include <bits/stdc++.h>
using namespace std;
bool isPrime(int num)
{
    for (int i=2; i<num; i++)
        if (num % i == 0) {
            return false; // check that no is prime or not.
        }
    return true;
}
```

```
int main()
{
    int a, b;
    cin >> a >> b;
```

```
for (int i=a; i<=b; i++)
    if (isPrime(i))
        cout << i;
```

// Here we learn how to send multiple numbers in fun.

2) Fibonacci Sequence

0, 1, 1, 2, 3, 5, 8

```
#include <bits/stdc++.h>
using namespace std;
```

```
void Fib(int num){
```

```
    int t1=0;
```

```
    int t2=1;
```

```
    int nextteam;
```

```
    for (int i=1; i<=num; i++) {
```

```
        cout << t1 << endl;
```

```
        nextteam = t1 + t2;
```

```
        t1 = t2;
```

```
        t2 = nextteam;
```

```
}
```

```
    return;
```

// first team 0
// second team 1
// next = first + second

then

t1 = second.

t2 = next.

```
int main() {
```

```
    int n;
```

```
    cin >> n;
```

```
    Fib(n);
```

```
}
```

Q) Find an Factorial of a number n.

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

include < bits/stdc++.h>
using namespace std;

int Fact (int num) {

int ans = 1;

for (int i=1; i<=num; i++) { // 2 se jese bhi
ans = ans * i;

}

return ans;

}

int main() {

int n;

cin >> n;

int ans = Fact(n);

cout << ans;

}

* Calculate ${}^n C_r$. (Binary Coefficient)

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

include < bits/stdc++.h>

using namespace std;

int fact (int num) {

int ans = 1;

for (int i=1; i<=num; i++) {

ans = ans * i;

... These are only repeated
Factorial finding.

}
return ans;

int main() {

int n, r;

cin >> n >> r;

int n1 = Fact(n);

int e1 = Fact(r);

int diff = Fact(n - r);

int ans = n1 / e1 * diff;

cout << ans << endl;

}

* Pascale Triangle.

~~1
2
3
4
5
6
7
8~~

$$t_{i,j} = \frac{i!}{j!}$$

$$\frac{i!}{j!(i-j)!}$$

~~$$nCr = \frac{n!}{r!(n-r)!}$$~~

1 0 0

1 1 0 1 1 0 1

1 2 1 2 1 2 1

1 3 1 3 1 3 1 3

1 4 1 4 1 4 1 4 1

#include <bits/stdc++.h>
using namespace std;

int fact (num);

int ans = 1;

for (int i = 1; i <= num; i++) {

ans = ans * i;

}
return ans;

```
int main() {
```

```
    int n;
```

```
    cin >> n;
```

```
    for (int i = 0; i < n; i++)
```

```
        for (int j = 0; j <= i; j++)
```

```
            cout << fact(i) / (fact(j) * fact(i - j))
```

```
    }
```

```
    }
```