Time Table { FJP - 2 }

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
9 PM class	9PM class	9PM class	off	9PM class	Weekly Test 9PM	Test + Doubts Discussion

Continuation of class

① Getting
② Patterns

Level ①

- Functions & Arrays
- 2D Arrays
- String & StringBuilder

DSA

- Recursion & Backtracking
- Searching & Sorting
- Dynamic Programming

My Introduction

→ Pointers

→ Pepcoding Articles (30 - 40%)

→ Job-Switch program - 4 { 3-4 months }

→ leetcode

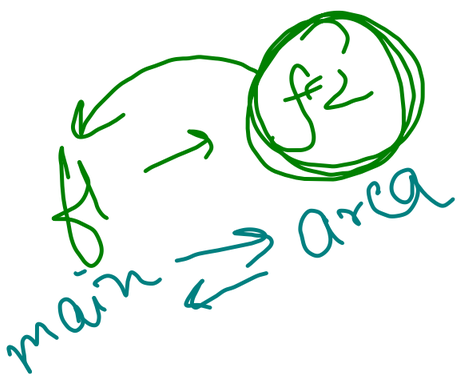
→ linkedIn

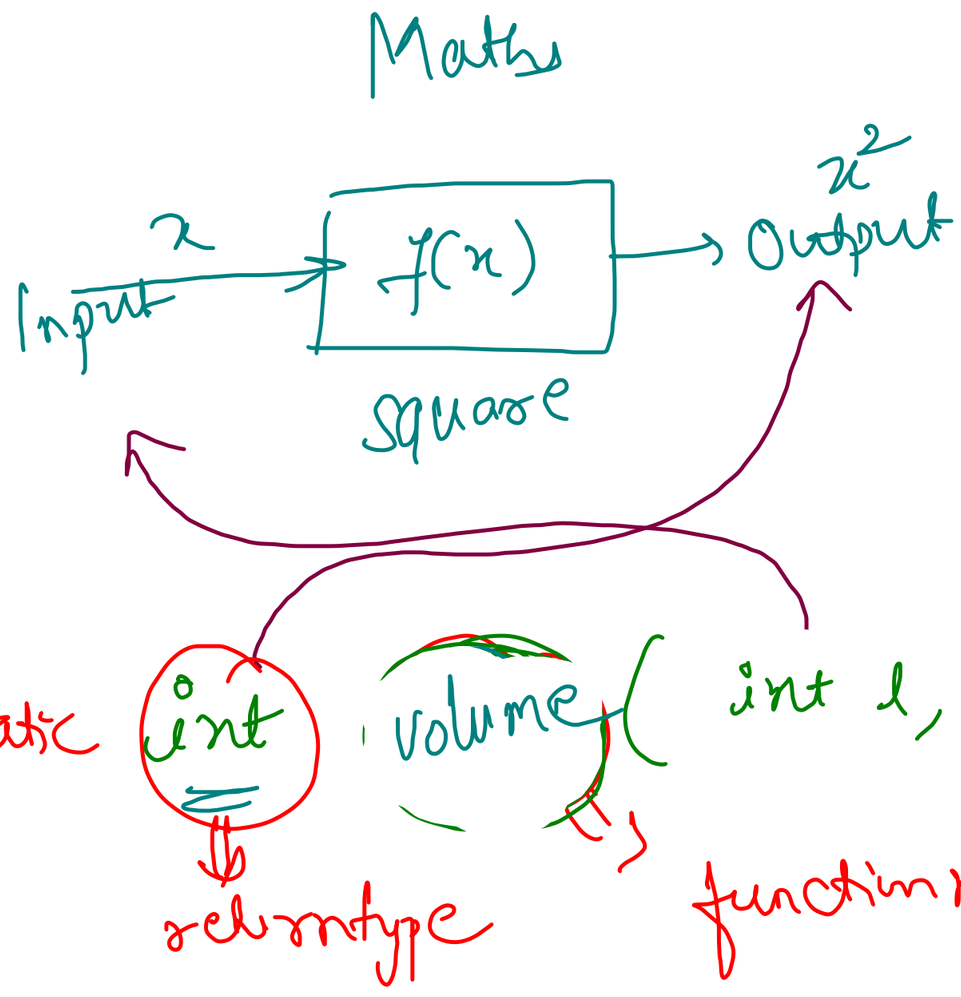
→ Competitive Programming

→ Product Based company

- Google (3)
- Salesforce (AMTS) (2)
- Sprinklr (1)

Functions

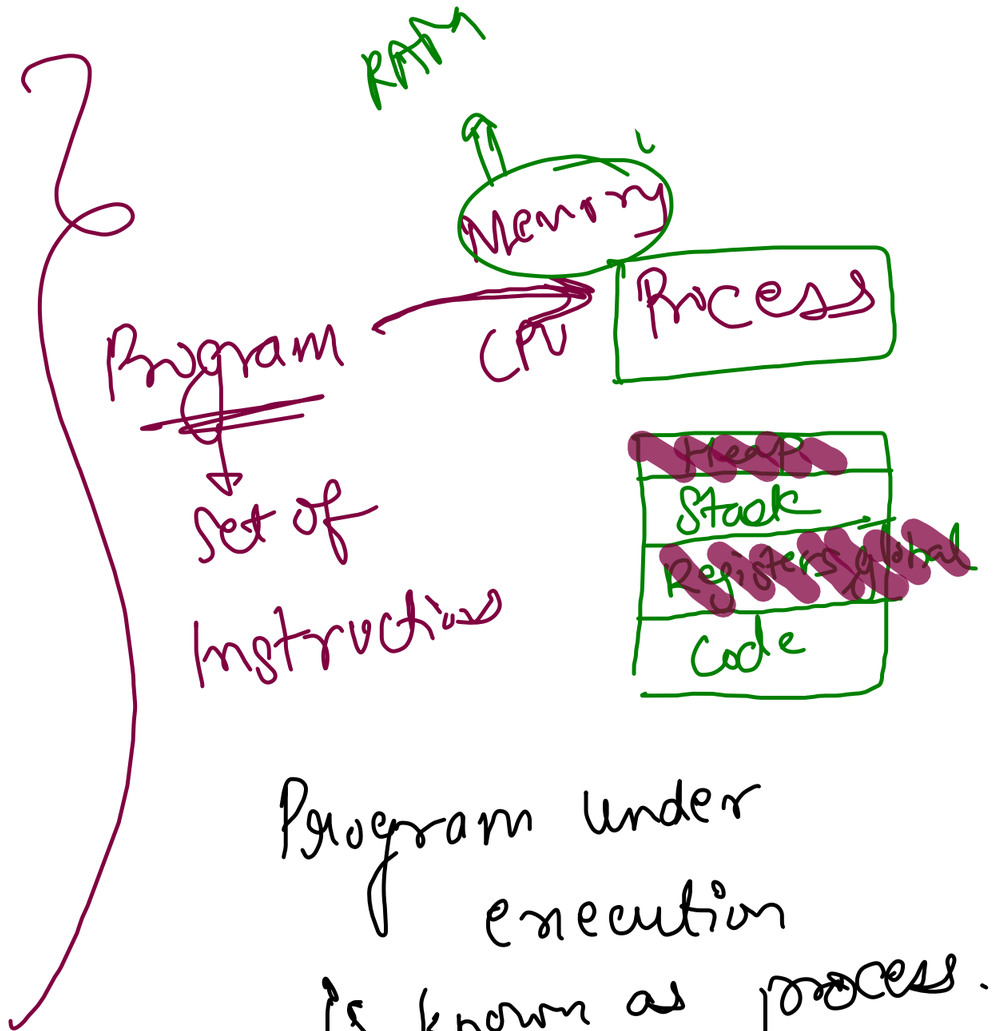





```
public static int volume( int l, int b, int h) {

}
```

returntype

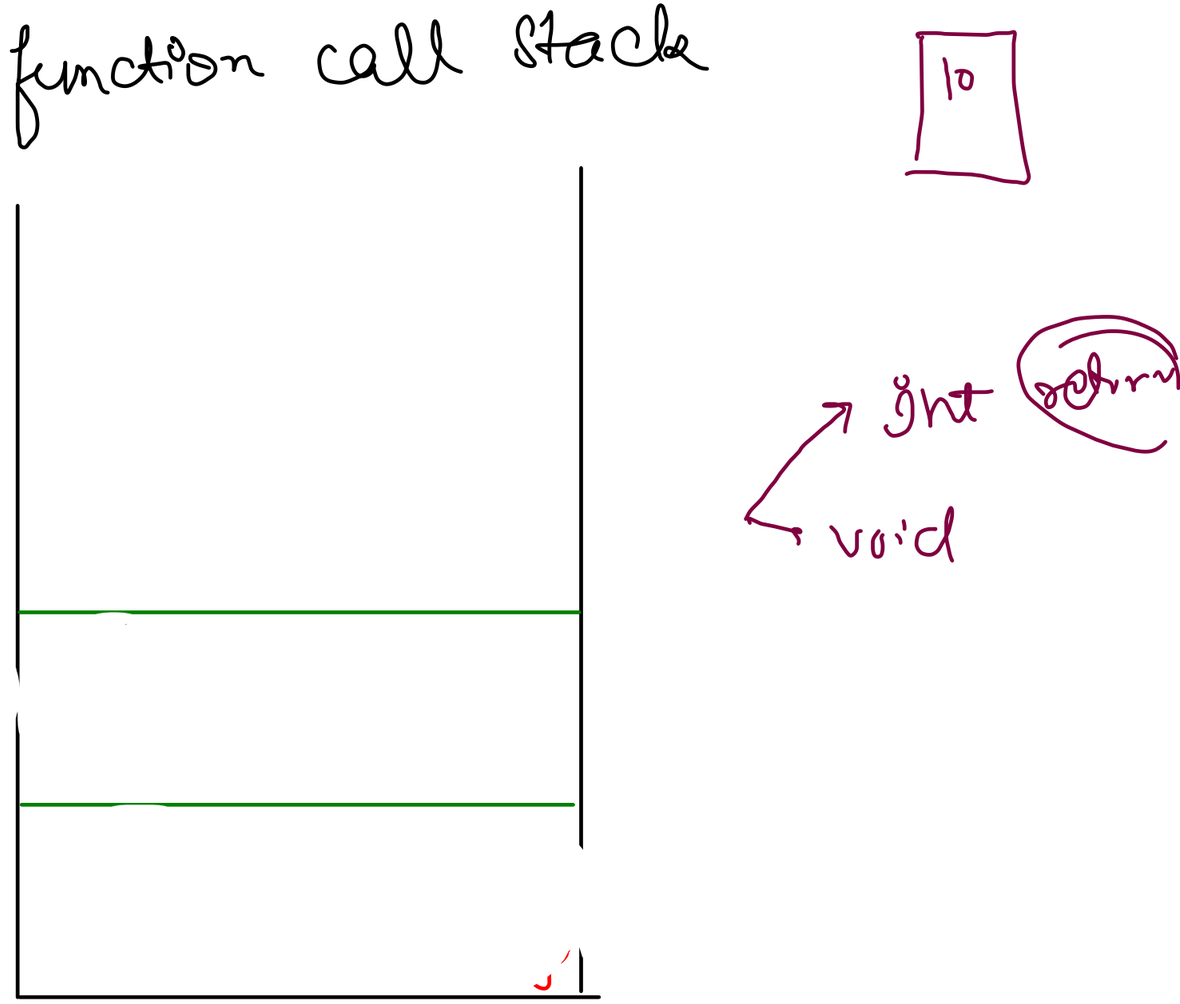
functionname



OS

Program under execution is known as process.

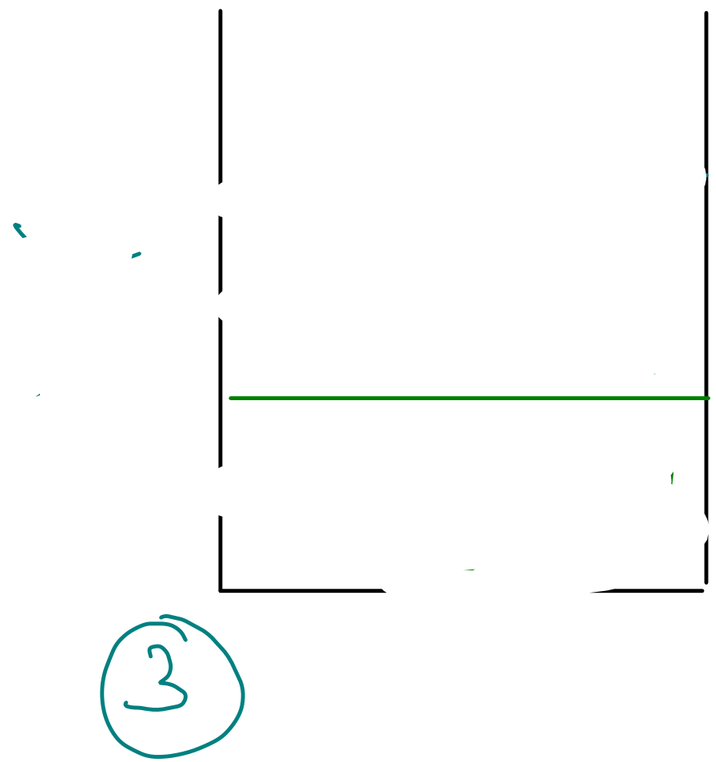
```java
public static int factorial(int n){
    int res = 1;

    for(int i=1; i<=n; i++){
        res = res * i;
    }

    return res;
}

public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);

    int n = scn.nextInt();
    int r = scn.nextInt();


    int nfact = factorial(n);
    int rfact = factorial(r);
    int nmrfact = factorial(n - r);


    int ncr = (nfact / (rfact * nmrfact));
    System.out.println(ncr);
}
```



n = 1292126429

d = 2

freq = 0 1 2 3



3

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int d = scn.nextInt();
    int f = getDigitFrequency(n, d);
    System.out.println(f);
}


public static int getDigitFrequency(int n, int d){

    int freq = 0;

    // For Extracting Digits
    while(n > 0){
        int digit = n % 10;

        if(digit == d){
            freq++;
        }

        n = n / 10;
    }

    return freq;
}
```

Number System

Decimal No

0-9

10 digits

$10 = 1*10^1 + 0*10^0$

100 > 99

$100 =$

0, 1, 2, 3, 4, ... 9

10, 11, 12, ... 20

21, 22, ... 30

..... 99

100

251693

$= 6*10^2 + 9*10^1 + 3*10^0$
$+ 1*10^3 + 5*10^4$
$+ 2*10^5$

0-9 0-9 0-9 0-9

- min no 0000
- max no 9999
- count of no's 10^4

Binary No

2

0 1

00
01
10
11

0
1

000
001
010
011
1100
101
110
111

2^2 2^1 2^0

$1101011 = 1*2^0 + 1*2^1 + 0*2^2 + 1*2^3 + 0*2^4 + ---$

$2*2*2*2$

0/1 0/1 0/1 0/1

min 0000

max 1111

Count 2^4

Octal Nb System

→ 0, 1, 2, 3, 4, 5, 6, 7

0 1 2 3 4 5 6 7

10 11 12 13 14 15 16 17

20 21 ... 27

30 31 ... 37

40 50 60

47 57 67

70 77

100 101 102 107

777 1000 7777 10000

$(7\ 5\ 3\ 4\ 6\ 1\ 2)_8$

$= 2*8^0 + 1*8^1 + 6*8^2$

$+ 4*8^3 + 3*8^4 + 5*8^5 + 7*8^6$

$(777)_8$ = $7*8^2 + 7*8^1 + 7*8^0$

1000 = $1*8^3$

$(999)_{10} = 1000$

$9*10^2 + 9*10^1 + 9*10^0$

$1*10^3$

6 months + leetcode (L2)

→ leetcode

Octal

$8*8*8*8$

_ _ _ _

min → 0000

max → 7777

total → 8^4

pepcoding (L1)

L2 → gfg

Service based

tier 3

tier 1 (3rd, 4th)

(2-3) hr

8 hr

DSA

GFG 200

2) Dev

DSA { 600 + leetcode }

Dev { 50 + Project }

150 → backtrack (crossed out)

→ DSA

80-90

40-50