

→ while loop.

→ Practice session on loops

Number System

Decimal \rightarrow binary .
↓

base .

base \rightarrow decimal .

base \rightarrow base .



Functions :-

$$n! = n \times (n-1) \times (n-2) \times (n-3) \times \dots \times 1$$

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

$$\frac{n!}{(n-x)!} = \frac{n \times (n-1) \times (n-2) \times \dots \times 1}{x \times (x-1) \times \dots \times 1}$$

$$\frac{5!}{3! 2!} = \frac{5!}{3! 2!}$$

$$= 5^2 \times 2!$$

$$= 3! \times 2!$$

$$= 10$$

$$\text{int ans1} = 1; \quad n=5; \quad x=3$$

for (int i=1; i<=n; i++)
ans1 = ans1 * i;

$$ans1 = ans1 * i; \quad \Rightarrow \quad ans1 = 120$$

$$5! = 120$$

int ans2 = 1;

for (int i=1; i<=x; i++)
ans2 *= i;

$$ans2 *= i; \quad \Rightarrow \quad ans2 = 6$$

int ans3 = 1;

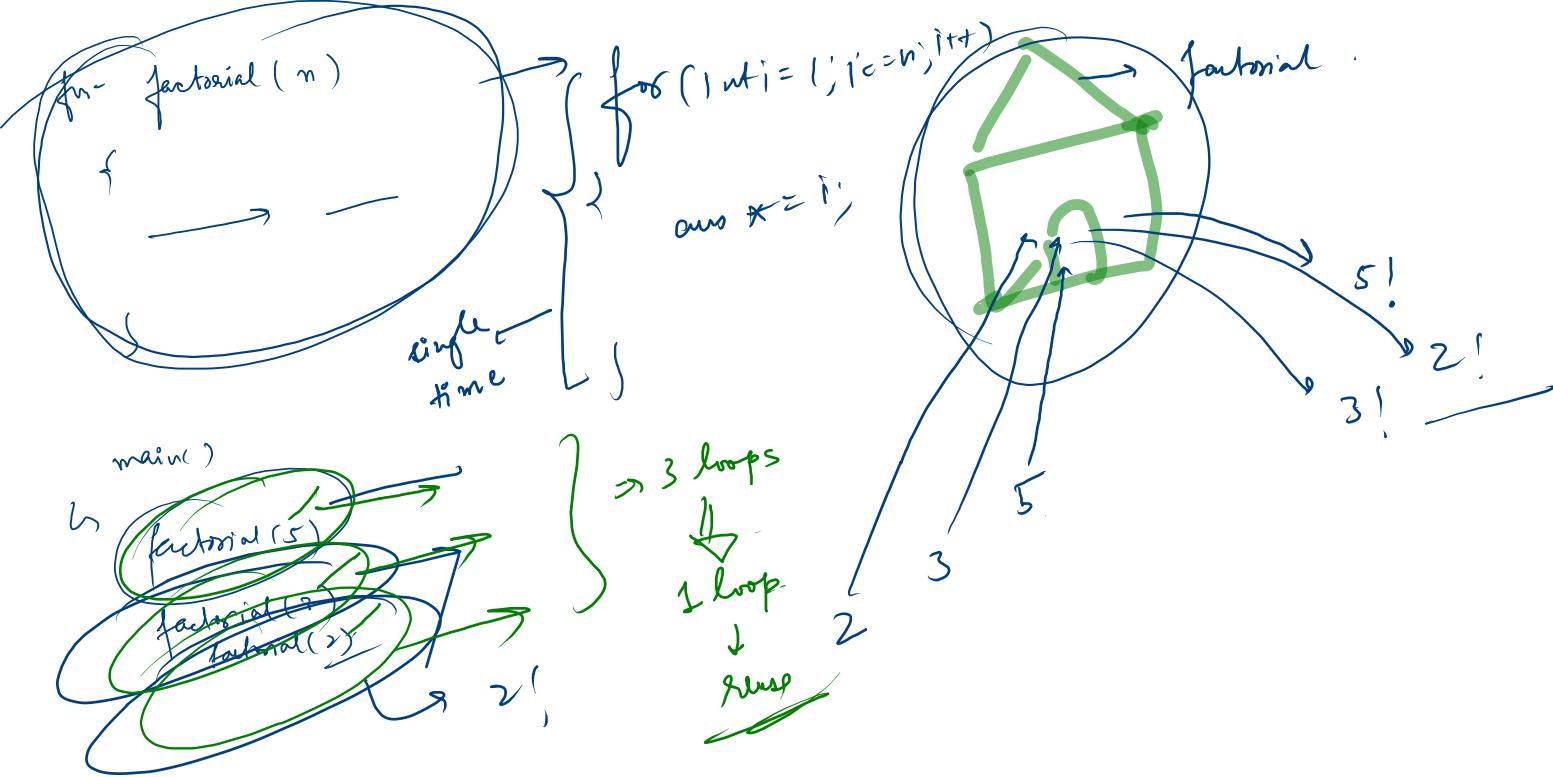
for (int i=1; i<=(n-x); i++)
ans3 *= i;

$$ans3 *= i; \quad \Rightarrow \quad (5-3)! = 2!$$

$$ans = 120 / (6 \times 2)$$

$$\Rightarrow ans = 10$$

DRY



Syntax :-

what type of value fn is going to return →

public static **return-type** function-name (optional) {

return)

integer \Rightarrow factorial \rightarrow integer $5! = 120$
string \Rightarrow name \rightarrow "Name"
boolean \Rightarrow true, false
return nothing

Factorial function :-

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

n!

Finding 5C_3 :-

(a) Without function

```

int ans1 = 1;
for(int i=1;i<=5;i++){
    ans1 *=i;
}

int ans2 = 1;
for(int i=1;i<=3;i++){
    ans2 *= i;
}

int ans3 = 1;
for(int i=1;i<=2;i++){
    ans3 *= i;
}

int ans = ans1/(ans2*ans3);
System.out.println(ans);

}

```

(b) With function

```

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

    return ans;
}

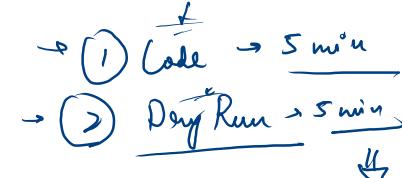
public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT. You
     */
    int ans1 = factorial(5);
    System.out.println(ans1);

    /*
    int ans2 = factorial(3);
    System.out.println(ans2);

    /*
    int ans3 = factorial(2);
    System.out.println(ans3);

    /*
    int ans = ans1/(ans2*ans3);
    System.out.println(ans);
}

```



```

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

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    int ans1 = factorial(5); → 120
    System.out.println(ans1); → 120

    int ans2 = factorial(3); → 6
    System.out.println(ans2);

    int ans3 = factorial(2); → 2
    System.out.println(ans3);

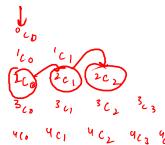
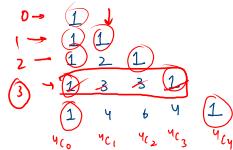
    int ans = ans1/(ans2*ans3);
    System.out.println(ans);
}

```

$\text{ans} = 1;$
 $i = 1 \leftarrow 5 \text{ (T)}$ $\Rightarrow \text{ans} = 1 * 1 = 1$
 $i = 2 \leftarrow 5 \text{ (T)}$ $\Rightarrow \text{ans} = 1 * 2 = 2$
 $i = 3 \leftarrow 5 \text{ (T)}$ $\Rightarrow \text{ans} = 2 * 3 = 6$
 $i = 4 \leftarrow 5 \text{ (T)}$ $\Rightarrow \text{ans} = 6 * 4 = 24$
 $i = 5 \leftarrow 5 \text{ (T)}$ $\Rightarrow \text{ans} = 24 * 5 = \underline{\underline{120}}$
 $i = 6 \leftarrow 5 \text{ (F)}$

Sample Output 0

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

 $\lambda = 0$ 

$$\frac{n_{C_{x+1}}}{n_{C_x}} = \frac{\frac{n!}{(x+1)! (n-x-1)!}}{\frac{n!}{x! (n-x)!}}$$

$$= \frac{\cancel{n!}}{(x+1) (x!) (n-x-1)!} \\ \cancel{n!} \\ \cancel{(x!) (n-x) (n-x-1)!}$$

$$\Rightarrow \frac{n_{C_{x+1}}}{n_{C_x}} = \frac{(n-x)}{(x+1)}$$

$$1_{C_{x+1}} = \frac{(1-x)*1}{1}$$

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

$$n_{C_x} \& n_{C_{x+1}} = \frac{1+x}{1} + 1 = 5+4 = 9$$

$$\frac{n!}{x! (n-x)!}, \quad \frac{n!}{(x+1)! (n-x-1)!}$$

$$\Rightarrow n_{C_{x+1}} = \frac{(n-x)}{(x+1)} [n_x]$$

$$i=3 \rightarrow 3c_0$$

$$\eta_{c_{r+1}} = \frac{\eta_{c_r(n-r)}}{(r+1)}$$

val

$$i_{(j+1)} = \frac{i c_i (i-j)}{(j+1)}$$

Java 8

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void pattern(int n){
        for(int i=0;i<n;i++){
            int val = 1;
            for(int j=0;j<=i;j++){
                System.out.print(val + " ");
                val = (val*(i-j))/(j+1);
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        pattern(n);
    }
}
```

Swans

Find Frequency(17 june)

Problem

Submissions

Leaderboard

Discussions

1. You are given a number n .

2. You are given a digit d .

3. You are required to calculate the frequency of digit d in number n .

Input Format

A number n A digit d

Constraints

$0 \leq n \leq 10^9$ $0 \leq d \leq 9$

Output Format

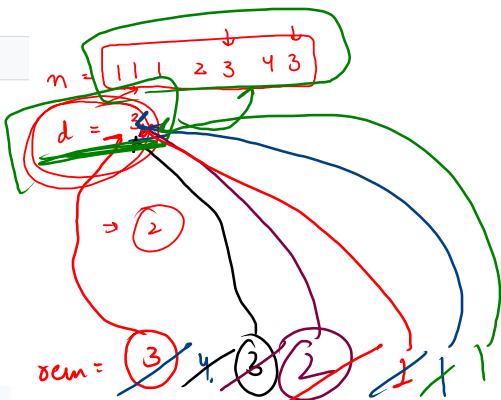
A number representing frequency of digit d in number n

Sample Input 0

$n \rightarrow 1112343$
 $d \rightarrow 3$

Sample Output 0

2



$$\boxed{1 \ 1 \ 1 \ 2 \ 3 \ 4} \ 3 \% / 10$$
$$\downarrow / 10$$

$$1 \ 1 \ 1 \ 2 \ 3 \ 4 \% / 10 = 4$$
$$\downarrow / 10$$

$$1 \ 1 \ 1 \ 2 \ 3 \% / 10 = 3$$
$$\downarrow / 10$$

$$1 \ 1 \ 1 \ 2 \% / 10 = 2$$
$$\downarrow / 10$$

$$1 \ 1 \ 1 \% / 10 = 1$$
$$\downarrow / 10$$

$$1 \% / 10 = 1$$
$$\downarrow / 10$$

$$1 \% / 10 = 1$$
$$\downarrow / 10$$
$$0$$

```
| import java.io.*;
| import java.util.*;
|
1 public class Solution {
2     public static int countGivenDigit(int n, int d){
3         int count = 0;
4         while(n>0){
5             int rem = n%10;
6             if(rem == d){
7                 count++;
8             }
9             n = n/10;
10        }
11
12        return count;
13    }
14
15    public static void main(String[] args) {
16        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
17        Scanner scn = new Scanner(System.in);
18        int n = scn.nextInt();
19        int d = scn.nextInt();
20
21        // int ans = countGivenDigit(n,d);
22        // System.out.println(ans);
23
24        System.out.println(countGivenDigit(n,d));
25    }
26}
```

→ Code
→ Dry Run