

Zero-One Triangle Pattern

i) Problem Statement

This problem is to understand the nested loop. Given N, a Positive integer, You are supposed to print the alternating 1's and 0's in triangle format.

Input Format :

Input is positive integer : 5

Output Format:

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

Code :

```
import java.util.Scanner;

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

        int n = sc.nextInt();

        for(int i=0; i<n; i++) {
            for(int j=0; j<i+1; j++) {
                System.out.print((i+j+1)%2 + " ");
            }
            System.out.println();
        }
    }
}
```

Output :

```

D:\230701298>javac oneZeroPattern.java
D:\230701298>java oneZeroPattern
5
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1

```

ii) Number-increasing reverse Pyramid Pattern

Given N, a Positive integer, You are supposed to print in the below format.

Sample Input:

6

Sample Output:

```

1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1

```

Code :

```

import java.util.Scanner;

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

        int n = sc.nextInt();

        for(int i=n-1; i>=0; i--) {
            int num = n - (n-1);
            for(int j=i+1; j>0; j--) {
                System.out.print(num + " ");
                num++;
            }
            System.out.println();
        }
    }
}

```

Output :

```
D:\230701298>javac numPyramid.java
D:\230701298>java numPyramid
6
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Identify the Weekday or Weekend

Problem Statement:

SYNTAX OF SWITCH CASE

The general syntax for a switch case in Java is as follows:

```
switch (expression) {
```

```
case value1:
```

```
// Code to be executed if expression equals value1
```

```
break;
```

```
case value2:
```

```
// Code to be executed if expression equals value2
```

```
break;
```

```
// ...
```

```
default:
```

```
// Code to be executed if expression doesn't match any case values
```

```
}
```

You are developing a scheduling application where users can check whether a given day is a weekday or a weekend. The application should prompt the user to enter a day of the week (e.g., "Monday", "Saturday"), and based on the input, the program should determine if the day is a weekday or a weekend.

Input Format

Input consists a week of the day

Output Format

Print whether it is weekday or weekend or invalid day

Sample Input 1

Monday

Sample Output 1

It's a weekday

Sample Input 2

Sunday

Sample Output 2

It's a weekend

Code :

```
import java.util.Scanner;

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

        String day = sc.nextLine();
        String d = "None";

        switch(day) {
            case("Sunday"):
            case("Saturday"):
                d = "It's a weekend";
                break;
            case("Monday"):
            case("Tuesday"):
            case("Wednesday"):
            case("Thursday"):
            case("Friday"):
                d = "It's a weekday";
                break;
            default :
                d = "It's not a valid day !";
                break;
        }
        System.out.println(d);
    }
}
```

Output :

```
D:\230701298>javac weekSchedule.java
D:\230701298>java weekSchedule
Sunday
It's a weekend
D:\230701298>javac weekSchedule.java
D:\230701298>java weekSchedule
Monday
It's a weekday
D:\230701298>javac weekSchedule.java
D:\230701298>java weekSchedule
january
It's not a valid day !
```

Strong Number

Problem Statement:

Write a program to check whether a number is a Strong Number or not.

A strong number is a positive integer whose sum of the factorials of its digits equals the original number

Few examples of strong numbers are : 1,2,145 and 40585.

Input Format:

Read the positive number

Output Format:

Print Whether it is strong number or not.

Sample Input 1:

145

Sample Output 1:

Strong number

Code :

```
import java.util.Scanner;
```

```

public class StrongNum {

    public static int factorial(int num) {

        int factorial = 1;

        for(int i = 1; i<num; i++) factorial += factorial*i;
        return factorial;

    }

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

        int num = sc.nextInt();
        int n = num;
        int sum = 0;

        while(num !=0 ) {
            int rem = num % 10;
            sum += factorial(rem);
            num /= 10;
        }
        if(sum == n)    System.out.println("Strong number");

        else    System.out.println("Not a strong number");
    }
}

```

Output :

```
D:\230701298>javac StrongNum.java
```

```
D:\230701298>java StrongNum
```

```
2
```

```
Strong number
```

```
D:\230701298>javac StrongNum.java
```

```
D:\230701298>java StrongNum
```

```
145
```

```
Strong number
```

```
D:\230701298>javac StrongNum.java
```

```
D:\230701298>java StrongNum
```

```
40585
```

```
Strong number
```

```
D:\230701298>javac StrongNum.java
```

```
D:\230701298>java StrongNum
```

```
25
```

```
Not a strong number
```