Week 2

1) i) Problem Statement

This problem 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 AlternatingTriangle {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a positive integer: ");

int N = scanner.nextInt();

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

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

if ((i + j) % 2 == 0) {

System.out.print("1 ");

} else {

System.out.print("0 ");

}

}

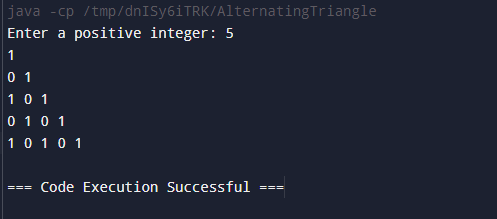
System.out.println();

}

}

}

OUTPUT:



2) 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 NumberReversePyramid {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a positive integer: ");

int N = scanner.nextInt();

for (int i = N; i >= 1; i--) {

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

System.out.print(j + " ");

}

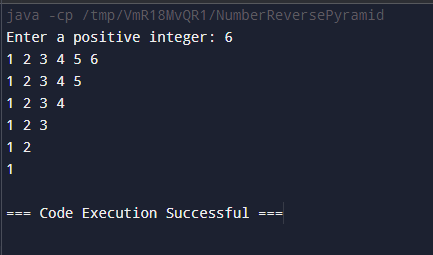
System.out.println();

}

}

}

OUTPUT:



3) 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&#39;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., &quot;Monday&quot;, &quot;Saturday&quot;), 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 DayChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a day of the week: ");

String day = scanner.nextLine().trim();

switch (day.toLowerCase()) {

case "monday":

case "tuesday":

case "wednesday":

case "thursday":

case "friday":

System.out.println("Its a weekday");

break;

case "saturday":

case "sunday":

System.out.println("Its a weekend");

break;

default:

System.out.println("Invalid day");

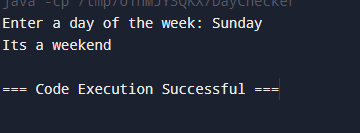
break;

}

}

}

OUTPUT:



4) 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 StrongNumber {

public static int factorial(int n) {

int fact = 1;

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

fact \*= i;

}

return fact;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a positive number: ");

int number = scanner.nextInt();

int originalNumber = number;

int sum = 0;

while (number > 0) {

int digit = number % 10;

sum += factorial(digit);

number /= 10;

}

if (sum == originalNumber) {

System.out.println("Strong number");

} else {

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

}

}

}

OUTPUT:

