**Coding Assignments Programs**

**Program 1**

Even Number: A number which is divisible by 2 and generates a remainder of 0 is called an even number. All the numbers ending with 0, 2, 4, 6, and 8 are even numbers.

Odd Number: An number that is not divisible by 2 and generates a remainder of 1 is called an odd number. All the numbers ending with 1, 3, 5,7, and 9 are odd numbers.

Generic Illustration for any random integer, check whether it is even or odd

Input : 13

Output: ODD

Input : 24

Output: EVEN

Program to check given number is even or odd

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n;

n=sc.nextInt();

if(n%2==0)

{

System.out.println("Even Number");

}

else

{

System.out.println("Odd Number");

}

}

}

**Program 2**

Program to print biggest of five numbers

import java.util.\*;

public class Biigest

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int a,b,c,d,e;

a=sc.nextInt();

b=sc.nextInt();

c=sc.nextInt();

d=sc.nextInt();

e=sc.nextInt();

if (a>b && a>c && a>d && a>e)

{

System.out.println(a+" is big");

}

else if (b>a && b>c && b>d && b>e)

{

System.out.println(b+" is big");

}

else if (c>a && c>b && c>d && c>e)

{

System.out.println(c+" is big");

}

else if (d>a && d>c && d>b && d>e)

{

System.out.println(d+" is big");

}

else

{

System.out.println(e+" is big");

}

}

}

**Program 3**

Program to print biggest of four numbers.

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int a,b,c,d;

a=sc.nextInt();

b=sc.nextInt();

c=sc.nextInt();

d=sc.nextInt();

if (a>b && a>c && a>d)

{

System.out.println(a+" is big");

}

else if (b>a && b>c && b>d )

{

System.out.println(b+" is big");

}

else if (c>a && c>b && c>d)

{

System.out.println(c+" is big");

}

else

{

System.out.println(d+" is big");

}

}

}

**Program 4**

Program to read a character and print “vowel” or “consonant” using nested-if

Import java.util.\*;

class Char

{

public static void main(String[ ] arg)

{

int i=0;

Scanner sc=new Scanner(System.in);

System.out.println("Enter a character : ");

char ch=sc.next( ).charAt(0);

if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'||ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U')

{

System.out.println("Entered character "+ch+" is  Vowel");

}

else if((ch>='a'&&ch<='z')||(ch>='A'&&ch<='Z'))

System.out.println("Entered character "+ch+" is Consonant");

      else

System.out.println("Not an alphabet");

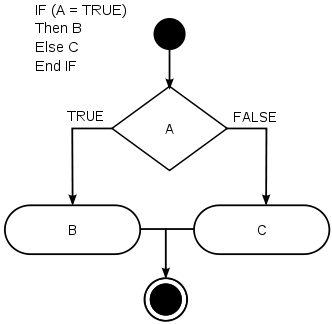
}

}

**Program 5**

<https://www.hackerrank.com/challenges/java-if-else/problem> ( nested- if hacker rank)

In this challenge, we test your knowledge of using *if-else* conditional statements to automate decision-making processes. An if-else statement has the following logical flow:



Source:[Wikipedia](https://en.wikipedia.org/wiki/Conditional_%28computer_programming%29)

**Task**  
Given an integer, , perform the following conditional actions:

* If  is odd, print Weird
* If  is even and in the inclusive range of  to , print Not Weird
* If  is even and in the inclusive range of  to , print Weird
* If  is even and greater than , print Not Weird

Complete the stub code provided in your editor to print whether or not  is weird.

**Input Format**

A single line containing a positive integer, .

**Constraints**

**Output Format**

Print Weird if the number is weird; otherwise, print Not Weird.

**Sample Input 0**

3

**Sample Output 0**

Weird

**Sample Input 1**

24

**Sample Output 1**

Not Weird

**Explanation**

*Sample Case 0:*   
 is odd and odd numbers are weird, so we print Weird.

*Sample Case 1:*   
 and  is even, so it isn't weird. Thus, we print Not Weird.

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n;

n=sc.nextInt();

if(n%2==0)

{

System.out.println("Not Weird ");

}

else

{

System.out.println("Weird ");

}

}

}

**Program 6**

Program on to read a number and print it is leap year or not (nested –if)

**Programs on switch:**

class Leapyear

{

public static void main(String arg[])

{

    long year;

                 Scanner sc=new Scanner(System.in);

    System.out.print("enter  any calendar year :");

                   year=sc.nextLong();

if(year!=0)

{

   if(year%400==0)

                   System.out.println(year+" is a leap year");

                   else  if(year%100==0)

     System.out.println(year+" is not a leap year");

                   else if(year%4==0)

     System.out.println(year+" is a leap year");

                       else

   System.out.println(year+" is not a leap year");

}

else

System.out.println("Year zero does not exist ");

}

}

**Program 7**

**Programs on switch:**

Program to print days of a week using switch case.

Import java.util.\*;

class WeekDays  
{  
    public static void main(String s[])  
    {

Scanner sc=new Scanner(System.in);  
        int day;

day=sc.nextInt();  
          
        switch(day)  
        {  
            case 1:  
                System.out.println("Monday");  
                break;  
            case 2:  
                System.out.println("Tuesday");  
                break;  
            case 3:  
                System.out.println("Wednesday");  
                break;  
            case 4:  
                System.out.println("Thursday");  
                break;  
            case 5:  
                System.out.println("Friday");  
                break;  
            default:  
                System.out.println("Weekend");  
                break;  
        }  
    }  
}

**Program 8**

Program to print no.of. days in a month by reading month number.

Import java.util.\*;

public class DaysinMonth2 {

private static Scanner sc;

public static void main(String[] args)

{

int month;

sc = new Scanner(System.in);

System.out.print(" Please Enter Month Number from 1 to 12 (1 = Jan, and 12 = Dec) : ");

month = sc.nextInt();

switch(month)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

System.out.println("\n 31 Days in this Month");

break;

case 4:

case 6:

case 9:

case 11:

System.out.println("\n 30 Days in this Month");

break;

case 2:

System.out.println("\n Either 28 or 29 Days in this Month");

break;

default:

System.out.println("\n Please enter Valid Number between 1 to 12");

}

}

}

**Program 9**

Program to check given character is vowel or consonant

import java.util.\*;

class SwitchVowel

{

public static void main()

{

Scanner sc=new Scanner(System.in);

System.out.print("\n Enter Character: ");

char c=((sc.nextLine()).charAt(0));

char z=Character.toUpperCase(c); //Changing Value to UpperCase for uniformity.

switch(z) //Checking Vowel Character using Switch Case

{

case 'A':

case 'E':

case 'I':

case 'O':

case 'U': System.out.println(c+" is a Vowel.");

break;

default: System.out.println(c+" is a Non-Vowel Character.");

}

}

}

**Program 10**

Write a C program to print all even numbers from 1 to n using for loop. C program to generate all even numbers between given range. Logic to print even numbers using if else and for loop in given range in C programming.

**Example**

**Input**

Input upper range: 10

**Output**

Even numbers between 1 to 10:

2, 4, 6, 8, 10

Solution

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int i=1;

while(i<=n)

{

if(i%2==0)

System.out.print(i+” “);

i++;

}

}

}

**Program 11**

Program to print range(n to m) of even numbers using while loop

https://www.geeksforgeeks.org/python-program-to-print-all-even-numbers-in-a-range/

Given starting and end points, write a Java program to print all even numbers in that given range.

Example:

Input: start = 4, end = 15

Output: 4, 6, 8, 10, 12, 14

Input: start = 8, end = 11

Output: 8, 10

Solution:

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int start=sc.nextInt();

int end=sc.nextInt();

while(start<=end)

{

if(start%2==0){

System.out.println(start+" ");

}

start++;

}

}

}

**Program 12**

Program to print range of even(n to m) numbers in reverse order using while loop.

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int start=sc.nextInt();

int end=sc.nextInt();

while(end>=start)

{

if(end%2==0){

System.out.println(end+" ");

}

end--;

}

}

}

**Program 13**

Program to print multiplication table upto 12 using for loop

Given a number n as input, we need to print its table, where N>0.

**Example**

**Input 1 :**- N = 7

**Output :**-

7 \* 1 = 7

7 \* 2 = 14

7 \* 3 = 21

7 \* 4 = 28

7 \* 5 = 35

7 \* 6 = 42

7 \* 7 = 49

7 \* 8 = 56

7 \* 9 = 63

7 \* 10 = 70

class GFG {

    public static void main(String[] args)

    {

Scanner sc=new Scanner(System.in);

int N=sc.nextInt();

                for (int i = 1; i <= 12; i++) {

            System.out.println(N + " \* " + i + " = "+ N \* i);

        }

    }

}

**Program 14**

Program to print multiplication table upto given number using for loop

class GFG {

    public static void main(String[] args)

    {

Scanner sc=new Scanner(System.in);

int N=sc.nextInt();

        int range = sc.nextInt();

        int i = 1;

        while (i <= range) {

             System.out.println(N + " \* " + i + " = "+ N \* i);

            i++;

        }

    }

}

**Program 15**

Program to read a number and print reverse of a number.

**Reversing**a number means that the digit at the first position should be swapped with the last digit, the second digit will be swapped with the second last digit, and so on till the middle element,

To reverse a number following steps should be performed:

* *Take the number’s modulo by 10*
* *Multiply the reverse number by 10 and add modulo value into the reverse number.*
* *Divide the number by 10.*
* *Repeat above steps until number becomes zero.*

import java.io.\*;

class GFG {

    public static void main (String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

      int rev = 0;

 int rem;

      while(n>0){

        rem = n%10;

        rev = (rev\*10) + rem;

        n = n/10;

      }

        System.out.print("Reversed Number is "+ reverse(n));

    }

}

**Program 16**

Program to print no.of digits in a number.

<https://codeforwin.org/2016/10/c-program-to-count-number-of-digits-in-number.html>

Write a C program to input a number from user and count number of digits in the given integer using loop. How to find total digits in a given integer using loop in C programming. Logic to count digits in a given integer without using loop in C program.

**Example**

**Input**

Input num: 35419

**Output**

Number of digits: 5

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int count=0;

while(n>0)

{

count++;

n=n/10;

}

System.out.println(count);

}

}

**Program 17**

<https://www.geeksforgeeks.org/count-even-odd-digits-integer/>

A certain number is given and the task is to count even digits and odd digits of the number and also even digits are present even a number of times and, similarly, for odd numbers.

**Print Yes If:**

If number contains even digits even number of time

Odd digits odd number of times

**Else**

Print No

**Examples :**

Input : 22233

Output : NO

count\_even\_digits = 3

count\_odd\_digits = 2

In this number even digits occur odd number of times and odd

digits occur even number of times so its print NO.

Input : 44555

Output : YES

count\_even\_digits = 2

count\_odd\_digits = 3

In this number even digits occur even number of times and odd

digits occur odd number of times so its print YES.

(Program to print no.of. even digits & odd digits in a number)

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int evencount=0,oddcount=0;

int rem;

while(n>0)

{

rem=n%10;

if(rem%2==0)

evencount++;

else

oddcount++;

n=n/10;

}

if(evencount%2==0 && oddcount%2==1)

System.out.println("YES");

else

System.out.println("NO");

}

}

**Program 17**

**Program to print sum of all digits for a given number , sum must be less than 10 i.e., sum(12345) = sum(15)=6.**

[**https://www.geeksforgeeks.org/finding-sum-of-digits-of-a-number-until-sum-becomes-single-digit/**](https://www.geeksforgeeks.org/finding-sum-of-digits-of-a-number-until-sum-becomes-single-digit/)

Finding sum of digits of a number until sum becomes single digit

* Difficulty Level : [Medium](https://www.geeksforgeeks.org/medium/)
* Last Updated : 26 Mar, 2021

Given a number **n**, we need to find the sum of its digits such that:

**Examples :**

Input : 1234

Output : 1

Explanation : The sum of 1+2+3+4 = 10,

digSum(x) == 10

Hence ans will be 1+0 = 1

**Input :** 5674

**Output :** 4

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int sum=0;

while (n > 0 || sum > 9)

{

if (n == 0) {

n = sum;

sum = 0;

}

sum += n % 10;

n /= 10;

}

System.out.println(sum);

}

}

**Program 18**

<https://www.geeksforgeeks.org/program-for-armstrong-numbers/>

Given a number x, determine whether the given number is Armstrong number or not. A positive integer of **n digits** is called an Armstrong number of **order n** (order is number of digits) if.

**abcd... = pow(a,n) + pow(b,n) + pow(c,n) + pow(d,n) + ....**

**Example:**

Input : 153

Output : Yes

153 is an Armstrong number.

1\*1\*1 + 5\*5\*5 + 3\*3\*3 = 153

The idea is to first count number digits (or find order). Let the number of digits be n. For every digit r in input number x, compute rn. If sum of all such values is equal to n, then return true, else false.

import java.util.\*;

public class Main

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int ams=0;

int rem;

int num=n;

while(n>0)

{

rem=n%10;

ams+=rem\*rem\*rem;

n=n/10;

}

if(ams==num)

System.out.println("True");

else

System.out.println("False");

}

}

**Program 19**

Program to print given number is **disarium number** or not. Ex: 135-🡪 1^1+3^2+5^3=135 🡪 135 disarium number

https://www.w3resource.com/java-exercises/numbers/java-number-exercise-11.php

Write a Java program to check whether a given number is a Disarium number or unhappy number.

A Disarium number is a number defined by the following process:  
Sum of its digits powered with their respective position is equal to the original number.  
For example 175 is a Disarium number:  
As 11+32+53 = 135  
Some other DISARIUM are 89, 175, 518 etc.  
A number will be called Disarium if the sum of its digits powered with their respective position is equal with the number itself. Sample Input: 135.

import java.util.Scanner;

public class Example11 {

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.print("Input a number : ");

int num = sc.nextInt();

int copy = num, d = 0, sum = 0;

String s = Integer.toString(num);

int len = s.length();

while(copy>0)

{

d = copy % 10;

sum = sum + (int)Math.pow(d,len);

len--;

copy = copy / 10;

}

if(sum == num)

System.out.println("Disarium Number.");

else

System.out.println("Not a Disarium Number.");

}

}

**Program 20**

Program to print number is **pronic number** or not.

The numbers that can be arranged to form a rectangle are called[Rectangular Numbers (also known as Pronic numbers)](https://www.geeksforgeeks.org/rectangular-numbers/). The first few Pronic numbers are:   
0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 182, 210, 240, 272, 306, 342, 380, 420, 462 . . . . . .   
Pronic number is a number which is the product of two consecutive integers, that is, a number n is a product of x and (x+1). The task is to check and print Pronic Numbers in a range.  
**Examples :** 

Input : 6

Output : Pronic Number

Explanation: 6 = 2 \* 3 i.e 6 is a product

of two consecutive integers 2 and 3.

Input :56

Output :Pronic Number

Explanation: 56 = 7 \* 8 i.e 56 is a product

of two consecutive integers 7 and 8.

Input : 8

Output : Not a Pronic Number

Explanation: 8 = 2 \* 4 i.e 8 is a product of

2 and 4 which are not consecutive integers.

import java.io.\*;

import java.util.\*;

import java.math.\*;

class GFG

{

    // function to check Pronic Number

    static boolean checkPronic(int x)

    {

        for (int i = 0;

             i <= (int)(Math.sqrt(x));

             i++)

            // Checking Pronic Number

            // by multiplying consecutive

            // numbers

            if (x == i \* (i + 1))

                return true;

        return false;

    }

    // Driver Code

    public static void main(String[] args)

    {

        // Printing Pronic

        // Numbers upto 200

        for (int i = 0; i <= 200; i++)

            if (checkPronic(i))

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

    }

}

**Program 21**

Program on **happy number:**

<https://leetcode.com/problems/happy-number/>

<https://www.javatpoint.com/program-to-check-happy-number>

n this program, we need to determine whether a given number is a happy number or not.

**Happy number**

The happy number can be defined as a number which will yield 1 when it is replaced by the sum of the square of its digits repeatedly. If this process results in an endless cycle of numbers containing 4, then the number is called an unhappy number.

For example, 32 is a happy number as the process yields 1 as follows

32 + 22 = 13

12 + 32 = 10

12 + 02 = 1

Some of the other examples of happy numbers are 7, 28, 100, 320 and so on.

**public** **class** HappyNumber

{

    //isHappyNumber() will determine whether a number is happy or not

**public** **static** **int** isHappyNumber(**int** num){

**int** rem = 0, sum = 0;

        //Calculates the sum of squares of digits

**while**(num > 0){

            rem = num%10;

            sum = sum + (rem\*rem);

            num = num/10;

        }

**return** sum;

    }

**public** **static** **void** main(String[] args) {

**int** num = 82;

**int** result = num;

**while**(result != 1 && result != 4){

            result = isHappyNumber(result);

        }

        //Happy number always ends with 1

**if**(result == 1)

            System.out.println(num + " is a happy number");

       //Unhappy number ends in a cycle of repeating numbers which contains 4

**else** **if**(result == 4)

            System.out.println(num + " is not a happy number");

    }

}

**Program 22**

Program to print given number is **harshad number** or not Ex: 156=1+5+6=12🡪 156/12=0 🡪 156 harshad number

https://www.geeksforgeeks.org/harshad-or-niven-number/

Harshad (Or Niven) Number

* Difficulty Level : [Easy](https://www.geeksforgeeks.org/easy/)
* Last Updated : 17 May, 2021

An integer number in base 10 which is divisible by the sum of its digits is said to be a Harshad Number. An *n-harshad* number is an integer number divisible by the sum of its digit in base n.  
Below are the first few Harshad Numbers represented in base 10:  
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 18, 20………  
Given a number in base 10, our task is to check if it is a Harshad Number or not.

**Examples :**

**Input:** 3

**Output:** 3 is a Harshad Number

**Input:** 18

**Output:** 18 is a Harshad Number

**Input:** 15

**Output:** 15 is not a Harshad Number

public class GFG {

    public static void main(String[] args)

    {

Scannner sc=new Scanner(System.in);

int n=sc.nextInt();

        int sum = 0;

        for (int temp = n; temp > 0; temp /= 10)

            sum += temp % 10;

         if (n % sum == 0)

System.out.println(n+” is a Harshad Number”);

else

Syatem.out.println(n+” is not a Harshad Number”);

}

}

**Program 23**

Program to find given number is perfect number or not

<https://www.geeksforgeeks.org/perfect-number/>

A number is a perfect number if is equal to sum of its proper divisors, that is, sum of its positive divisors excluding the number itself. Write a function to check if a given number is perfect or not.   
**Examples:**

Input: n = 15

Output: false

Divisors of 15 are 1, 3 and 5. Sum of

divisors is 9 which is not equal to 15.

Input: n = 6

Output: true

Divisors of 6 are 1, 2 and 3. Sum of

divisors is 6.

Solution:

**import** java.util.Scanner;

**public** **class** PerfectNumberExample1

{

**public** **static** **void** main(String args[])

{

**long** n, sum=0;

Scanner sc=**new** Scanner(System.in);

n=sc.nextLong();

**int** i=1;

**while**(i <= n/2)

{

**if**(n % i == 0)

{

Sum = sum + i;

}

i++;

}

**if**(sum==n)

{

System.out.println(n+" is a perfect number.");

}

**else**

System.out.println(n+" is not a perfect number.");

}

}

**Program 24**

Automorphic number:

<https://www.geeksforgeeks.org/automorphic-number/>

Given a number N, the task is to check whether the number is Automorphic number or not. A number is called Automorphic number if and only if its square ends in the same digits as the number itself.  
**Examples :** 

Input : N = 76

Output : Automorphic

Explanation: As 76\*76 = 57**76**

Input : N = 25

Output : Automorphic

As 25\*25 = 6**25**

Input : N = 7

Output : Not Automorphic

As 7\*7 = 49

**public** **class** AutomorphicNumberExample1

{

**public** **static** **void** main(String args[])

{

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

**int** square = num \* num;

**while** (num > 0)

{

**if** (num % 10 != square % 10) {

**System.out.println(“Not Automorphic”);**

break;

}

num = num/10;

square = square/10;

}

System.out.println(“Automorphic”);

}

}

Solution two:

import java.util.Scanner;

public class Example14 {

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.print("Input a number : ");

int num = sc.nextInt();

int sq\_num = num\*num;

String str\_num = Integer.toString(num);

String square = Integer.toString(sq\_num);

if(square.endsWith(str\_num))

System.out.println("Automorphic Number.");

else

System.out.println("Not an Automorphic Number.");

}

}

**Program 25**

**Program on different arguments in for loop.**

Program on for with multiple arguments: to print the following output. i 🡪 5; j🡪 50.

1 5

2 4

3 3

4 2

5 1

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

for(int i=1,j=num;i<=num && j>0;i++,j--)

System.out.println(i+" "+j);

}

}

**Program 26**

Nested for loops

Program to generate multiplication chart.

public **class** MulTableInJava{

public static **void** main (String args[]){

int MulTable[][]=**new** int[10][10];

int row=1,column=1;

**for**(int i=0; i<MulTable.length; i++){

**for**(int j=0; j<MulTable[i].length; j++){

MulTable[i][j]=row\*column;

column=column+1;

}

row=row+1;

column=1;

}

**for**(int i=0; i<MulTable.length; i++){

**for**(int j=0; j<MulTable[i].length; j++){

System.out.print(" "+MulTable[i][j]+"\t ");

}

System.out.print("\n");

}

}

}

**Program 27**

Program to print following pattern:

\*

\* \*

\* \* \*

\* \* \* \*

Solution:

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

for(int i=0;i<num;i++)

{

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

{

System.out.print("\* ");

}

System.out.println("\n");

}

}

}

**Program 28**

Print the following pattern

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

Solution

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

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

{

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

{

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

}

System.out.println("\n");

}

}

}

**Program 29**

Print the following pattern

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

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

{

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

{

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

}

System.out.println("\n");

}

}

}

**Program 30**

Print the following pattern

A

B C

D E F

G H I J

K L M N O

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

char ch='A';

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

{

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

{

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

ch++;

}

System.out.println("\n");

}

}

}

**Program 31**

Print the following pattern

A

B B

C C C

D D D D

E E E E E

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

char ch='A';

for(int i=1;i<=num;i++,ch++)

{

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

{

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

}

System.out.println("\n");

}

}

}

**Program 32**

Print the following pattern

A

A B

A B C

A B C D

A B C D E

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

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

{

char ch='A';

for(int j=1;j<=i;j++,ch++)

{

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

}

System.out.println("\n");

}

}

}

**Program 33**

Print the following pattern

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

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

{

for(int k=num-i;k>0;k--)

System.out.print(" ");

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

{

System.out.print("\*");

}

System.out.println("\n");

}

}

}

**Program 34**

Print the following pattern

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

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

{

for(int k=num-i;k>0;k--)

System.out.print(" ");

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

{

System.out.print("\* ");

}

System.out.println("\n");

}

}

}

**Program 35**

Print the following pattern

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

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

{

for(int k=num-i;k>0;k--)

System.out.print(" ");

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

{

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

}

System.out.println("\n");

}

}

}

**Program 36**

Print the following pattern

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

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

{

for(int k=num-i;k>0;k--)

System.out.print(" ");

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

{

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

}

System.out.println("\n");

}

}

}

**Program 37**

Print the following pattern

A

A B

A B C

A B C D

A B C D E

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

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

{

for(int k=num-i;k>0;k--)

System.out.print(" ");

char ch='A';

for(int j=1;j<=i;j++,ch++)

{

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

}

System.out.println("\n");

}

}

}

**Program 38**

Print the following pattern

A

B B

C C C

D D D D

E E E E E

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

char ch='A';

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

{

for(int k=num-i;k>0;k--)

System.out.print(" ");

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

{

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

}

ch++;

System.out.println("\n");

}

}

}

**Program 39**

Print the following patterns

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

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

{

for(int j=1;j<=num;j++)

{

System.out.print("\*");

}

System.out.println("\n");

}

}

}

**Program 40**

Print the following patterns Solid Rhobus

<https://www.geeksforgeeks.org/program-to-print-solid-and-hollow-rhombus-patterns/>

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

import java.util.\*;

public class MyClass {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int num=sc.nextInt();

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

{

for(int k=num-i;k>0;k--)

System.out.print(" ");

for(int j=1;j<=num;j++)

{

System.out.print("\*");

}

System.out.println("\n");

}

}

}