1. Given an input integer, you must determine which primitive data types are capable of

properly storing that input.

Input Format

The first line contains an integer, , denoting the number of test cases.

Each test case, , is comprised of a single line with an integer, , which can be arbitrarily large or

small.

Output Format

For each input variable and appropriate primitive , you must determine if the given primitives

are capable of storing it. If yes, then print: n can be fitted in:\* dataType

If there is more than one appropriate data type, print each one on its own line and order them by

size (i.e.: ).

If the number cannot be stored in one of the four aforementioned primitives, print the line:

n can&#39;t be fitted anywhere.

Sample Input

5

-150

150000

pgm1:

import java.util.\*;

public class weeks

{ public static void main(String args[])

{ int n,i; long a;

Scanner get = new Scanner(System.in);

n=get.nextInt();

for(i=0;i<n;i++)

{

a=get.nextLong();

if(-128 <= a && a <= 128)

{

System.out.println("\* byte\n\* short\n\* int\n\* long");

}

else if( -32768 <= a && a <= 32767)

{ System.out.println("\* short\n\* int\n\* long");

}

else if(-2147483648 <= a && a <= 2147483647)

{

System.out.println("\* int\n\* long");

}

else if(Long. MIN\_VALUE <= a && a <= Long. MAX\_VALUE) { System.out.println("\* long");

}

else

{ System.out.println("Can't be fitted anywhere");

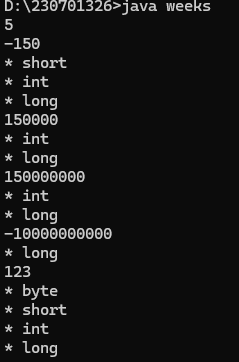
}

}

}

}

output:



2. You are developing a financial application that needs to handle both whole numbers and

decimal values. The application takes user inputs as integers (e.g., representing amounts in cents)

and needs to convert them to double for further calculations (e.g., converting cents to dollars).

The application should:

1. Take an integer amount in cents as input.

2. Convert this integer to a double to represent the amount in dollars.

3. Ensure that the conversion is accurate and the output is properly formatted to two decimal

pgm2:

import java.io.\*;

import java.util.Scanner;

public class pgm

{

public static void main(String[] args)

{

Scanner scan= new Scanner(System.in);

int cents=scan.nextInt();

double dollar=cents/100.00;

System.out.format(“The output be: %.2f”,dollar);

}

}

Output:



3. In a game, the player's score is calculated as a double value with high precision.

However, for display purposes, you need to show the score as an integer.

pgm3:

import java.util.\*;

public class weeks

{

public static void main(String[] args)

{

Scanner c=new Scanner(System.in);

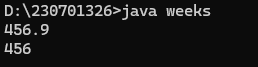
double a=c.nextDouble();

int b=(int)a;

System.out.println(b);

}

}



4. You are developing a payroll system where you need to calculate the adjusted salary

based on a percentage increase. The initial salary is given as an int, and the percentage increase

is given as a double.

Questions:

1. Input:

o Initial salary: 45000 (stored as int)

o Percentage increase: 7.5 (stored as double)

Output:

o Calculate the new salary after applying the percentage increase.

o Show how type promotion affects the calculation and what the resulting salary

would be.

Expected Output:

o The new salary after a 7.5% increase should be 48375.0 (as a double).

2. Input:

o Another initial salary: 32000 (stored as int)

o Percentage increase: 12.3 (stored as double)

Output:

o Calculate the new salary and discuss how type promotion is applied in the

calculation.

Expected Output: The new salary after a 12.3% increase should be 35976.0 (as a

double).

Pgm 4:

import java.util.\*;

public class pgm

{

public static void main(String[] args) {

Scanner c=new Scanner(System.in);

int a=c.nextInt();

double b=c.nextDouble();

double d=a\*(b/100);

double e=a+d;

System.out.println(e);

}

}

Output:



5. A mobile application for a puzzle game requires players to reverse

the digits of a given number to form a new number. The goal is to check if the

reversed number is equal to the original number.

Task: Write a Java program that reads an integer and reverses its digits. Check if

the reversed number is the same as the original.

import java.util.\*;

public class pgm

{

public static void main(String[] args) {

Scanner c=new Scanner(System.in);

int a=c.nextInt();

int temp=a;

int rev=0;

while(a!=0)

{

int rem= a%10;

rev=rev\*10+rem;

a=a/10;

}

System.out.println(rev);

if(rev==temp)

{

System.out.println("The reversed number is "+rev+". It is the same as the original.");

}

else{

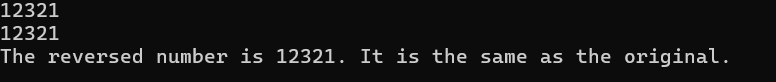
System.out.println("The reversed number is "+rev+". It is the not same as the original.");

}

}

}

Output:



6.A graphics tool allows users to create complex shapes for designs.

One of the patterns you need to implement is a diamond shape using stars (\*).

The user provides the number of rows in the top half of the diamond.

Task: Write a Java program that takes an integer n and prints a diamond pattern.

Pgm :

import java.io.\*;

import java.util.\*;

public class pgm

{

public static void main(String[] args)

{

Scanner num=new Scanner(System.in);

System.out.println("Enter valuee:");

int number=num.nextInt();

int m, n;

for (m = 1; m <= number; m++)

{

for (n = 1; n <= number - m; n++)

{

System.out.print(" ");

}

for (n = 1; n <= m \* 2 - 1; n++)

{

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

}

System.out.println();

}

for (m = number - 1; m > 0; m--)

{

for (n = 1; n <= number - m; n++)

{

System.out.print(" ");

}

for (n = 1; n <= m \* 2 - 1; n++)

{

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

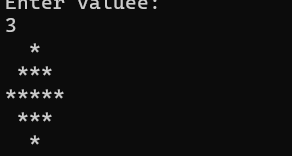
}

System.out.println();

}

}

Output:



7..You are developing a software for an advanced math visualization

tool. One of the features is to generate complex patterns that combine

mathematical concepts with visual representations. Specifically, you need to

create a pattern that combines Pascal's Triangle and a half-diamond shape.

Pgm:

import java.io.\*;

import java.util.\*;

public class pgm {

public static void Pascal(int n)

{

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

for (int j = 0; j <= n-i; j++) {

System.out.print(" ");

}

int a = 1;

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

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

a = a\* (i - k) / k;

}

System.out.println();

}

for (int i =n-1; i>0; i--) {

for (int j = 0; j <= n - i; j++) {

System.out.print(" ");

}

int a= 1;

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

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

a = a \* (i - k) / k;

}

System.out.println();

}

}

public static void main(String[] args)

{

Scanner num=new Scanner(System.in);

System.out.println("Enter Value:");

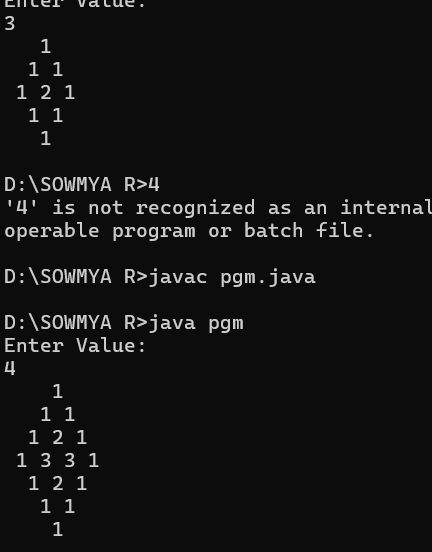
int n =num.nextInt();

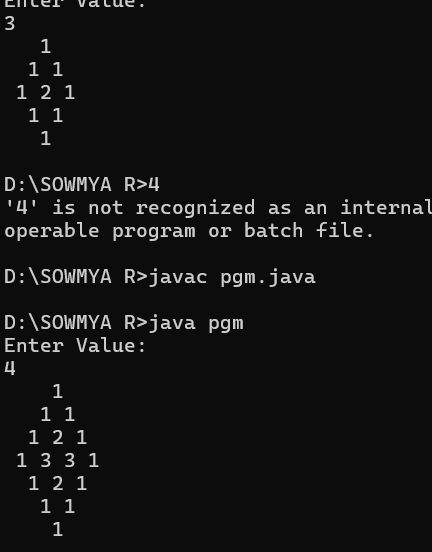
Pascal(n);

}

}

Output:



z

8. We use the integers a, b, and n to create the following series: (a+20 .b), (a+20 .b+21 .b),…,(a+20 .b+21 .b+…..+2n-1 .b) You are given q queries in the form of a, b, and n. For each query, print the series corresponding to the given a, b, and n values as a single line of n space\_separated integers.

Pgm:

import java.util.Scanner;

import java.io.\*;

public class pgm

{

public static void main(String[] args)

{

Scanner scan=new Scanner(System.in);

int q=scan.nextInt();

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

{

int a=scan.nextInt();

int b=scan.nextInt();

int n=scan.nextInt();

int sum=a;

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

{

sum+=(int) Math.pow(2,j)\*b;

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

}

System.out.println();

}

}

}

Output:

