

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

Code:

```
import java.util.Scanner;

class Type{

    public static void main(String[] args){

        Scanner ip=new Scanner(System.in);

        int n=ip.nextLong();

        if(n>=-128 && n<=127){

            System.out.println("byte");

        }

        if(n>=-32768 && n<=3276){

            System.out.println("short");

        }

        if(n>=-21474266486 && n<=21474266485){

            System.out.println("int");

        }

        if(n>=-92233720368547758086 && n<=92233720368547758086){

            System.out.println("long");

        }

    }

}
```

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).

Code:

```
import java.util.Scanner;

class Convert{
```

```

public static void main(String[] args){
    Scanner a=new Scanner(System.in);
    double c=a.nextInt();
    double d=c/100;
    System.out.println(d);
}
}

```

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.

Code:

```

import java.util.Scanner;
class Type{
    public static void main(String[] args){
        Scanner a=new Scanner(System.in);
        double c=a.nextDouble();
        int d=(int)c;
        System.out.println(d);
    }
}

```

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.

Code:

```

import java.util.Scanner;
class Salary{
    public static void main(String[] args){
        Scanner obj=new Scanner(System.in);

```

```

        int n=obj.nextInt();

        double p=obj.nextInt();

        double t=n+(n*(p/100));

        System.out.println(t);
    }
}

```

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.

Code:

```

import java.util.Scanner;

class Reverse{

    public static void main(String[] args){

        Scanner obj=new Scanner(System.in);

        int a=obj.nextInt();

        int rev=0;

        if(a>0){

            int r=a%10;

            rev=(rev*10)+r;

            a=a%10;

        }

        System.out.println(rev);

    }

}

```

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.

Code:

```
import java.util.*;

class Diam {

    public static void main(String[] args) {

        Scanner ip = new Scanner(System.in);

        int a = ip.nextInt();

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

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

                System.out.print(" ");

            }

            for (int k = 1; k <= (2*i-1) ; k++ ) {

                System.out.print("*");

            }

            System.out.println();

        }

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

            for (int j = a; j > i; j--) {

                System.out.print(" ");

            }

            for (int k = 1; k <= (2*i-1) ; k++ ) {

                System.out.print("*");

            }

            System.out.println();

        }

    }

}
```

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.

Code:

```
import java.util.Scanner;

class Pascal {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter an integer n: ");

        int n = scanner.nextInt();

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

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

                System.out.print(" ");

            }

            int num = 1;

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

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

                num = num * (i - j) / (j + 1);

            }

            System.out.println();

        }

        for (int i = n - 2; i >= 0; i--) {

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

                System.out.print(" ");

            }

            int num = 1;

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

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

                num = num * (i - j) / (j + 1);

            }

            System.out.println();

        }

    }

}
```

8. We use the integers a, b, and n to create the following series:

$(a+20.b), (a+20.b+21.b), \dots, (a+20.b+21.b+\dots+2n-1.b)$

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.

Code:

```
import java.util.*;
import java.io.*;
class Series{
    public static void main(String args[]){
        Scanner in =new Scanner(System.in);
        int t=in.nextInt();
        for(i=0;i<t;i++){
            int a=in.nextInt();
            int b=in.nextInt();
            int n=in.nextInt();
            int s=a;
            for(j=0;j<n;j++){
                int x=(int)Math.pow(2,j);
                s=s+x*b;
                System.out.println(s+" ");
            }
            System.out.println("\n");
        }
        in.close();
    }
}
```