

Text

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

/\* Signature of four required functions is given in the code. You can create other functions as well if you need.

\*  Also you should submit your code even if you are not done with all the functions.

\*/

// Main used internally is shown here just for your reference.

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

        Scanner s = new Scanner(System.in);

        int n = s.nextInt();

        int degree1[] = new int[n];

        for(int i = 0; i < n; i++){

            degree1[i] = s.nextInt();

        }

        int coeff1[] = new int[n];

        for(int i = 0; i < n; i++){

            coeff1[i] = s.nextInt();

        }

        Polynomial first = new Polynomial();

        for(int i = 0; i < n; i++){

            first.setCoefficient(degree1[i],coeff1[i]);

        }

        n = s.nextInt();

        int degree2[] = new int[n];

        for(int i = 0; i < n; i++){

            degree2[i] = s.nextInt();

        }

         int coeff2[] = new int[n];

        for(int i = 0; i < n; i++){

            coeff2[i] = s.nextInt();

        }

        Polynomial second = new Polynomial();

        for(int i = 0; i < n; i++){

            second.setCoefficient(degree2[i],coeff2[i]);

        }

        int choice = s.nextInt();

        Polynomial result;

        switch(choice){

        // Add

        case 1:

             result = first.add(second);

            result.print();

            break;

        // Subtract

        case 2 :

             result = first.subtract(second);

            result.print();

            break;

        // Multiply

        case 3 :

             result = first.multiply(second);

            result.print();

            break;

        }

    }

  \*/

import java.util.Scanner;

import java.util.ArrayList;

public class Polynomial {

    int coefficient[]=new int[100];

    /\* This function sets coefficient for a particular degree value, if degree is not there in the polynomial

     \*  then corresponding term(with specified degree and value is added int the polynomial. If the degree

     \*  is already present in the polynomial then previous coefficient is replaced by

     \*  new coefficient value passed as function argument

    \*/

    public void setCoefficient(int degree, int coeff){

coefficient[degree]+=coeff;

    }

    // Prints all the terms(only terms with non zero coefficients are to be printed) in increasing order of degree.

    public void print(){

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

            if(coefficient[i]!=0){

                System.out.print(coefficient[i]+"x"+i+" ");

            }

}

    }

    // Adds two polynomials and returns a new polynomial which has result

    public Polynomial add(Polynomial p){

for(int i=0;i<this.coefficient.length;i++){

            for(int j=0;j<p.coefficient.length;j++){

                if(i==j && (this.coefficient[i]!=0 || p.coefficient[j]!=0)){

                    this.coefficient[i]+=p.coefficient[j];

                }

            }

        }

return this;

    }

    // Subtracts two polynomials and returns a new polynomial which has result

    public Polynomial subtract(Polynomial p){

            for(int i=0;i<this.coefficient.length;i++){

            for(int j=0;j<p.coefficient.length;j++){

                if(i==j && (this.coefficient[i]!=0 || p.coefficient[j]!=0)){

                    this.coefficient[i]-=p.coefficient[j];

                }

            }

        }

return this;

    }

    // Multiply two polynomials and returns a new polynomial which has result

    public Polynomial multiply(Polynomial p){

        Polynomial a =new Polynomial();

        for(int i=0;i<this.coefficient.length/2;i++){

            for(int j=0;j<p.coefficient.length/2;j++){

                a.coefficient[(i+j)]+=this.coefficient[i]\*p.coefficient[j];

            }

        }

        return a;

    }

}