1. Keerthi, a receptionist of Hotel Benzz Park rearranges the chairs in the reception daily. Suresh, manager of the same hotel writes a program to display the arrangement on the reception TV screen. Today, Keerthi arranged the chairs as shown in the sample output(like Z shape). Help Suresh to write a program to display the arrangement pattern.

Note : Each chair has a number and there is more than one chair with the same number.

```
1 2 3 4 5
   4
  3
 2
1 2 3 4 5
```

Input Format

Total number of chairs in first line

Output Format

Display Z pattern as shown in sample output

Sample Input

```
5
```

Sample Output

```
1 2 3 4 5
   4
  3
 2
1 2 3 4 5
```

```java
import java.util.*;
public class qn1 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int num;
        System.out.print("Enter your number :");
        Scanner in = new Scanner(System.in);
        num=in.nextInt();

        for(int i=1 ; i<=num;i++) {
            for(int j=1 ; j<=num;j++) {
                if(i==1 || i==num) {
                    System.out.print(j);
                }
                else {
                    if(i==num-j+1) {
                        System.out.print(j);
                    }
                    else {
                        System.out.print(" ");
                    }
                }
            }
            System.out.println("");
        }
    }

}
```

2. ABC public school posts a problem everyday in their main board. The question of the day is there is a two integer(Number(N) and a Digit(d)). Students have to form an expression or equation that contains only D and that expression evaluates to N. Allowed operators in expression are +, -, * and /. Find the minimum length expression that satisfy the condition above

and D can only appear in the expression at most 10(limit) times. Hence limit the values of N(Although the value of limit depends upon how far you want to go. But a large value of limit can take longer time for below approach).

Ex: If N=5 and d=9

Output : (9+9+9+9+9)/9

The digit 9 has been used by five operators(addition(4) + division(1)).

Input Format

Input the Number (N) and a Digit(d).

Output Format

Display the minimum length expression that satisfy the condition above and D can only appear in the expression at most 10(limit) times.

Constraints

0< N <= 100

0 < d<= 100

(No negative numbers are allowed)

(If can't able form an expression means, the output will be "Expression not found")

Sample Input

5

9

Sample Output

Expression: (9+9+9+9+9)/9

```java
import java.util.*;

public class qn2 {
    public static void main (String arg[]){
        int D , N;
        Scanner in = new Scanner(System.in);
        N=in.nextInt();
        D=in.nextInt();
        System.out.print("(");
        for(int i =1 ; i<N ;i++){
            System.out.print(D+"+");

        }
        System.out.print(D+")/"+D);
    }
}
```

3. Keerthi, a receptionist of Hotel Benzz Park rearranges the chairs in the reception daily. Suresh, manager of the same hotel writes a program to display the arrangement on the reception TV screen. Today, Keerthi arranged the chairs as shown in the sample output(like Z shape). Help Suresh to write a program to display the arrangement pattern.

Note : Each chair has a number and there is more than one chair with the same number.

```
1 2 3 4 5
   4
  3
 2
1 2 3 4 5
```

Input Format

Total number of chairs in first line

Output Format

Display Z pattern as shown in sample output

Sample Input

5

Sample Output

```
1 2 3 4 5
   4
  3
 2
1 2 3 4 5
```

```java
import java.util.*;

public class qn3 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int num;
        System.out.print("Enter your number :");
        Scanner in = new Scanner(System.in);
        num=in.nextInt();

        for(int i=1 ; i<=num;i++) {
            for(int j=1 ; j<=num;j++) {
                if(i==1 || i==num) {
                    System.out.print(j);
                }
                else {
                    if(i==num-j+1) {
                        System.out.print(j);
                    }
                    else {
                        System.out.print(" ");
                    }
                }
            }
            System.out.println("");
        }
    }

}
```

4. A version Management system (VMS) is a repository of files, often the files for the source code of computer programs, with monitored access. Every change made to the source is tracked, along with who made the change, why they made it, and references to problems fixed, or enhancements introduced, by the change.

In this problem we will consider a simplified model of a development project. Let's suppose that there are N source files in the project. All the source files are distinct and numbered from 1 to N.

A VMS which is used for maintaining the project contains two sequences of source files. The first sequence contains M source files that are ignored by the VMS. If a source file is not in the first sequence, then it's considered to be unignored. The second sequence contains K source files that are tracked by the VMS. If a source file is not in the second sequence, then it's considered to be untracked.

A source file can either be or not be in any of these two sequences. Your task is to calculate two values: the number of source files of the project, that are both tracked and ignored, and the number of source files of the project, that are both untracked and unignored.

Sample 1 Input

7 4 6

Input Format

The first line of the input contains three integers N, M and K denoting the number of source files in the project, the number of ignored source files and the number of tracked source files. Assume that the maximum value for N as 50.

The second line contains M distinct integers denoting the sequence A of ignored source files. The sequence is strictly increasing.

The third line contains K distinct integers denoting the sequence B of tracked source files. The sequence is strictly increasing.

Output Format

Output a single line containing two integers: the number of the source

1 4 6 7

1 2 3 4 6 7

Sample 1 Output

4 1

Sample 2 Input

4 2 2

1 4

3 4

Sample 2 Output

1 1

```java
import java.util.*;
public class qn4 {
    public static void main(String arg[]){
        int N,M,K;
        Scanner in = new Scanner (System.in);
        N=in.nextInt();
        M=in.nextInt();
        K=in.nextInt();
        ArrayList <Integer> alM = new  ArrayList<Integer>();
        ArrayList <Integer> alK = new  ArrayList<Integer>();
        Set <Integer> set = new HashSet<>();
        int n=0;
        for(int i=0; i<M;i++){
            n=in.nextInt();
            alM.add(n);
            set.add(n);
        }
        for(int i=0; i<K;i++){
            n=in.nextInt();
            alK.add(n);
            set.add(n);
        }
        ArrayList <Integer> al3 = new ArrayList<Integer>(alM);
        al3.retainAll(alK);
        int num1 , num2;
        num1=al3.size();
        num2=N-set.size();
        System.out.println(num1+" "+num2);

    }

}
```

5. Rani, a receptionist of Hotel SMS Grand rearranges the flower in the reception daily. Sidharth, manager of the same Hotel writes a program to display the flower arrangement on the reception TV screen. Today, Rani arranged the flowers as shown in the sample output. Help Sidharth to write a program to display the pattern.

```
*
***
*****
*******
*********
```

Input Format

Number of rows

Output Format

Pattern as shown in sample output

Sample Input

5

Sample Output

```
*
***
*****
*******
*********
```

```java
import java.util.*;

public class qn5 {
    public static void main(String arg[]){
        int num=0,s=1;
        Scanner in = new Scanner (System.in);
        num=in.nextInt();
        for(int i=0;i<num;i++){
            for(int j=0;j<s;j++){
                System.out.print("*");
            }
            System.out.print("\n");
            s+=2;
        }
    }
}
```

6. Nolan has been provided with two strings a and b. His task is to change string a such that all the common letters of a and b is to be deleted and the uncommon letters of a and b is to be joined. Help him finish it.

Note: If no changed is possible display -1.

Input Format

The first line of input consists of a string a. The next line consists of a string b.

Output Format

Display the required output.

Constraints

$1 <= |\text{Length of Strings}| <= 10^4$

Sample Input

aacdb

gafd

Sample Output

cbgf

Sample Input

abcs

cxzca

Sample Output

bsxz

Sample Input

zryzsguaxdqnsqovilnmnntvselkkktqgimcwyydbcdiqduzrfhrudosqgxejyeuktqlcuijlzrppzusqyvcpk
wypwmlpemtdvcu

szzyfzchsjxsrjaiccqdjltnllidsxyyifojwqhzepbpkoazzvftncmvremwalijjrdsffevmaeyxdgpsjppvaysi
kknfxkvkchb

Sample Output

Uuuuuuu

```
import java.util.*;
public class qn6 {
    public static void main(String arg[]){

        Scanner in = new Scanner(System.in);
        String str1,str2;
        str1=in.nextLine();
        str2=in.nextLine();
        List<String> myList1 = new ArrayList<String>(Arrays.asList(str1.split("")));
        List<String> myList2 = new ArrayList<String>(Arrays.asList(str2.split("")));
        String s1="" ,s2="";
        for(String a:myList1){
            if(!myList2.contains(a)){

                s1=s1+a;
            }
        }
        for(String a:myList2){
            if(!myList1.contains(a)){
                s2=s2+a;
            }
        }

        System.out.println(s1+s2);


    }

}
```

7. Write a Java program that the elements in the multiplicands and product consist of all the unique integer from 1 to 9 if Input is 7254 ==>product is 39*186 thus it contains all the numbers between 1-9.

Input Format
Given an integer N.

Output Format
If they contain unique numbers.

The first line of the output prints the number along with their multiplicands as a string.

The second line prints yes.

else

Print no.

Constraints
Integers only.

Sample 1 Input
7254

Sample 1 Output
725439186

yes

Sample 2 Input
1234

Sample 2 Output
no

```
import java.util.*;
public class qn7 {
    public boolean check(String s){
        char arr[]=s.toCharArray();
        String numbers="123456789";
        Arrays.sort(arr);
        s="";

        for(char a:arr){
            s+=a;
        }
        if(s.equals(numbers)){
            return true;
        }
        else
            return false;
    }
    public static void main (String arg[]){
        qn7 obj = new qn7();
        String num;
        Scanner in = new Scanner (System.in);
        String numbers="123456789";
        ArrayList <String> rnum = new ArrayList<>(Arrays.asList(numbers.split("")));
        num = in.nextLine();
        int input=Integer.parseInt(num);
        for(int i=0;i<num.length();i++){
            rnum.remove(Character.toString(num.charAt(i)));
        }
        int size=rnum.size();
        ArrayList<String> arl2 = new ArrayList<String>(rnum);
        String ot;
        boolean flag=true;
        int it=0;
        while(flag){
            ArrayList <String> temp = new ArrayList<>();
            for(String b:rnum){
                for(int i =0 ; i<arl2.size();i++){
                    if(b.length()+arl2.get(i).length() >size+1){
                        System.out.println("no");
                        flag=false;
                        return;
                    }
                    int num1=Integer.parseInt(b);
                    int num2=Integer.parseInt(arl2.get(i));
                    if(num1*num2 == input){
                        if(obj.check(num+arl2.get(i)+b)){
                            System.out.println(num+b+arl2.get(i));
                            System.out.println("\nyes");
                            flag=false;
                            return;
                        }
                        else{
```

```
                            temp.add(arl2.get(i)+b);
                        }
                    }
                    else{
                    temp.add(arl2.get(i)+b);

                    it++;
                }

                }

            }
            arl2.addAll(temp);
            rnum.addAll(temp);
            temp.clear();
        }
    }


}
```

8. Jennie is working in a bank and her daily task is to guide the customers to fill various bank challans. A part of this, customers need to write the numbers in word format. Jennie decided to write a program to read the number and print the number digit by digit in word format. Help Jennie to complete this task.

Input Format
Number

Output Format
Given number in word format

Sample 1 Input
659803

Sample 1 Output
six five nine eight zero three

```
import java.util.*;

public class qn8 {
    public String check(String num){
        switch(num){
            case "0":
                return "zero";
            case "1":
                return "one";
            case "2":
                return "two";
            case "3":
                return "three";
            case "4":
                return "four";
            case "5":
                return "five";
            case "6":
                return "six";
            case "7":
                return "seven";
            case "8":
                return "eight";
            case "9":
                return "nine";
            default:
                return "";
        }
```

```
    }

    public static void main(String arg[]){
        String str;
        qn8 obj = new qn8();
        Scanner in = new Scanner(System.in);
        str = in.nextLine();
        ArrayList<String> al = new ArrayList<>(Arrays.asList(str.split("")));
        for(String s: al){
            System.out.print(obj.check(s)+" ");
        }

    }
}
```

9. Peter is working in the Data analyst company. His team leader gave a task to Peter. The task is, peter wants to find the number of occurrence of a certain letter in the given string. Help him to complete this task.

Input Format

First line of Input contains the String **S**

Next line contains the Character **C**

Output Format

Output the number of occurrences of the given letter.

Constraints

String should contain only alphabets

Sample Input

madam

a

Sample Output

2

```
import java.util.*;
public class qn9 {
    public static void main(String arg[]){
        String input ;
        Scanner in = new Scanner (System.in);
        input=in.nextLine();
        Map<String,Integer> map = new HashMap<>();
        ArrayList<String> newarray = new ArrayList<>(Arrays.asList(input.split("")));
        for(String s:newarray){
            if(map.containsKey(s)){
                int num=map.get(s);
                map.replace(s, num, num+1);
            }
            else{
                map.put(s, 1);
            }
        }

        String key ;
        key = in.nextLine();
        System.out.println(map.get(key));

    }

}
```

10. A student called Aron is very naughty in school. He is always bored in class, and he is always making a chaos. The faculty wanted to cool him down and "gentle" him, so he has given him a complex mathematical problem.

The faculty gives Aron an arithmetic expression A, the integer P and M. Aron has to answer the following question: "What is the minimum non-negative value of variable x in expression A so that the remainder of dividing A with M is equal to P?". The solution will always exist.

Additionally, it will hold that, if we apply the rules of distribution on expression A, variable x will not multiply variable x

(formally, the expression is a polynomial of the first degree in variable x).

Interpretations of valid expressions: 5+x·(3+2), x+3·x+4·(5+3·(2+x−2·x)).

Interpretations of invalid expressions: 5·(3+x·(3+x)), x·(x+x·(1+x)).

Input Format

The first line of input contains the expression A. The second line of input contains two integers P and M. The arithmetic expression A will only consists of characters +, -, *, (,), x and digits from 0 to 9. The brackets will always be paired, the operators +, - and * will always be applied to exactly two values (there will not be an expression (−5) or (4+−5)) and all multiplications will be explicit (there will not be an expression 4(5) or 2(x)).

Output Format

Display the minimal non-negative value of variable x.

Constraints

- **$1<=|A|<=10^5$**
- **$0<=P<=M-1$**
- **$1<=M<=10^6$**

Sample Input

5+3+x

Sample Output

9 10

1

Sample Input

20+3+x

Sample Output

0 5

2

Sample Input

3*(x+(x+4)*5)

Sample Output

1 7

1

```
import java.util.*;

public class qn10 {
    public int check(String s,int x){
        ArrayList<String> eq = new ArrayList<>(Arrays.asList(s.split("")));
        ArrayList<String> stackop = new ArrayList<>();
        ArrayList<Integer> stackn = new ArrayList<>();
        ArrayList<String> operand = new ArrayList<String>(Arrays.asList("+-
*/()x".split("")));
        int n=0;
        for(char a: s.toCharArray()){
                try{
                    if(Character.toString(a).equals("x")){
                        stackn.add(x);
                    }
                    else{
                        n=Integer.parseInt(Character.toString(a));
```

```
                    stackn.add(n);
                }
            }catch(Exception e){
                if(operand.contains(Character.toString(a))){
                String op= Character.toString(a);

                if(stackop.size()==0){
                    stackop.add(op);
                }
                else if(op.equals("/")){
                    if((stackop.get(stackop.size()-1).equals("/"))){
                        int num=stackn.get(stackn.size()-2)/stackn.get(stackn.size()-1);
                        stackn.remove(stackn.size()-1);
                        stackn.remove(stackn.size()-1);
                        stackn.add(num);
                        stackop.remove(stackop.size()-1);
                        stackop.add(op);
                    }
                    else if((stackop.get(stackop.size()-1).equals("*"))){
                        int num=stackn.get(stackn.size()-2)*stackn.get(stackn.size()-1);
                        stackn.remove(stackn.size()-1);
                        stackn.remove(stackn.size()-1);
                        stackn.add(num);
                        stackop.remove(stackop.size()-1);
                        stackop.add(op);
                    }
                    else{
                        stackop.add(op);
                    }
                }
                else if(op.equals("*")){
                    if((stackop.get(stackop.size()-1).equals("*"))){
                        int num=stackn.get(stackn.size()-2)*stackn.get(stackn.size()-1);
                        stackn.remove(stackn.size()-1);
                        stackn.remove(stackn.size()-1);
                        stackn.add(num);
                        stackop.remove(stackop.size()-1);
                        stackop.add(op);
                    }
                    else if((stackop.get(stackop.size()-1).equals("/"))){

                        int num=stackn.get(stackn.size()-2)/stackn.get(stackn.size()-1);
                        stackn.remove(stackn.size()-1);
                        stackn.remove(stackn.size()-1);
                        stackn.add(num);
                        stackop.remove(stackop.size()-1);
                        stackop.add(op);
                    }

                    else{
                        stackop.add(op);
                    }
                }

                else if(op.equals("+")){
                    if((stackop.get(stackop.size()-1).equals("+"))){
                        int num=stackn.get(stackn.size()-2)+stackn.get(stackn.size()-1);
                        stackn.remove(stackn.size()-1);
                        stackn.remove(stackn.size()-1);
                        stackn.add(num);
                        stackop.remove(stackop.size()-1);
                        stackop.add(op);
                    }
                    else if((stackop.get(stackop.size()-1).equals("-"))){
```

```
                    int num=stackn.get(stackn.size()-2)-stackn.get(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.add(num);
                    stackop.remove(stackop.size()-1);
                    stackop.add(op);
                }
                else if((stackop.get(stackop.size()-1).equals("/"))){

                    int num=stackn.get(stackn.size()-2)/stackn.get(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.add(num);
                    stackop.remove(stackop.size()-1);
                    stackop.add(op);
                }

                else if((stackop.get(stackop.size()-1).equals("*"))){

                    int num=stackn.get(stackn.size()-2)*stackn.get(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.add(num);
                    stackop.remove(stackop.size()-1);
                    stackop.add(op);
                }


                else{
                    stackop.add(op);
                }
            }
            else if(op.equals("-")){
                if((stackop.get(stackop.size()-1).equals("+"))){
                    int num=stackn.get(stackn.size()-2)+stackn.get(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.add(num);
                    stackop.remove(stackop.size()-1);
                    stackop.add(op);
                }
                else if((stackop.get(stackop.size()-1).equals("-"))){

                    int num=stackn.get(stackn.size()-2)-stackn.get(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.add(num);
                    stackop.remove(stackop.size()-1);
                    stackop.add(op);
                }
                else if((stackop.get(stackop.size()-1).equals("/"))){

                    int num=stackn.get(stackn.size()-2)/stackn.get(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.add(num);
                    stackop.remove(stackop.size()-1);
                    stackop.add(op);
                }

                else if((stackop.get(stackop.size()-1).equals("*"))){

                    int num=stackn.get(stackn.size()-2)*stackn.get(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
```

```
                        stackn.add(num);
                        stackop.remove(stackop.size()-1);
                        stackop.add(op);
                    }


                    else{
                        stackop.add(op);
                    }
                }
                else if(op.equals("(")){
                    stackop.add(op);
                }
                else if(op.equals(")")){
                    while(!stackop.get(stackop.size()-1).equals("(")){
                        switch(stackop.get(stackop.size()-1)){
                            case "+":
                                n=stackn.get(stackn.size()-2)+stackn.get(stackn.size()-
1);
                                stackn.remove(stackn.size()-1);
                                stackn.remove(stackn.size()-1);
                                stackn.add(n);
                                break;
                            case "-":
                                n=stackn.get(stackn.size()-2)-stackn.get(stackn.size()-
1);
                                stackn.remove(stackn.size()-1);
                                stackn.remove(stackn.size()-1);
                                stackn.add(n);
                                break;
                            case "*":
                                n=stackn.get(stackn.size()-2)*stackn.get(stackn.size()-
1);
                                stackn.remove(stackn.size()-1);
                                stackn.remove(stackn.size()-1);
                                stackn.add(n);
                                break;
                            case "/":
                                n=stackn.get(stackn.size()-2)/stackn.get(stackn.size()-
1);
                                stackn.remove(stackn.size()-1);
                                stackn.remove(stackn.size()-1);
                                stackn.add(n);
                                break;
                        }
                        stackop.remove(stackop.size()-1);
                    }
                    stackop.remove(stackop.size()-1);

                }

            }
        }
    }
    while(stackop.size()!=0){
        switch(stackop.get(stackop.size()-1)){
            case "+":
                n=stackn.get(stackn.size()-2)+stackn.get(stackn.size()-1);
                stackn.remove(stackn.size()-1);
                stackn.remove(stackn.size()-1);
                stackn.add(n);
                break;
            case "-":
                n=stackn.get(stackn.size()-2)-stackn.get(stackn.size()-1);
                stackn.remove(stackn.size()-1);
```

```
                    stackn.remove(stackn.size()-1);
                    stackn.add(n);
                    break;
                case "*":
                    n=stackn.get(stackn.size()-2)*stackn.get(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.add(n);
                    break;
                case "/":
                    n=stackn.get(stackn.size()-2)/stackn.get(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.remove(stackn.size()-1);
                    stackn.add(n);
                    break;
                case ")":
                    break;
            }
            stackop.remove(stackop.size()-1);
        }
        return stackn.get(stackn.size()-1);

    }

    public static void main(String arg[]){
        qn10 obj = new qn10();
        boolean fg=true;
        int i=0;
        Scanner in = new Scanner(System.in);
        String eqn = in.nextLine();
        int r=in.nextInt();
        int d=in.nextInt();
        int result;
        while(fg){
            result=obj.check(eqn,i);
            if((result%d)==r ){
                System.out.println(i);
                fg=false;
                break;
            }
            i++;
        }
    }

}
```

11. Assume there are N persons and each person needs exactly one cab. For each person, you are given the start time and end time (both inclusive) during which that person will travel. Find the minimum number of cabs required.

Input Format
First line contains an integer, N(1,<=N<=10^5) denoting the number of persons. Next N lines contains 4 integersHH1,MM1,HH2,and MM2, (0<=HH1,HH2<=23) (0<=MM1,MM2<=59), denoting the start time (HH1:MM1) and end time (HH2:MM2). It is guaranteed that start and end time will not span midnight.

Output Format
Print the minimum number of cabs required.

Sample 1 Input
Sample 1 Output
6

1 0 2 0

16 0 21 30

9 30 13 0

21 30 22 30

21 30 22 30

12 0 12 30

3

```java
import java.util.*;
public class qn11 {
    public static int count=1;
    public boolean check(ArrayList<Integer> a1 ,ArrayList<Integer> a2 ){
        int A1=a1.get(0)*100+a1.get(1);
        int A2=a1.get(2)*100+a1.get(3);
        int B1=a2.get(0)*100+a2.get(1);
        int B2=a2.get(2)*100+a2.get(3);
        if((B1>=A1) && (B1<A2) ){
            return true;
        }
        else if((A1>=B1)&&(A1<B2)){
            return true;
        }
        else{
            return false;
        }

    }
    public static void main(String arg[]){
        qn11 obj = new qn11();
        Scanner in = new Scanner(System.in);
        int num=in.nextInt();
        ArrayList<ArrayList> list = new ArrayList<>();
        for(int i=0 ; i<num;i++){
            ArrayList<Integer> numal = new ArrayList<>();

            int k;
            for(int j=0 ;j<4;j++){
                k=in.nextInt();
                numal.add(k);
            }
            if(list.size()!=0){
                for(ArrayList l :list){
                    if(obj.check(l, numal)){
                        qn11.count++;
                    }
                }
            }
            list.add(numal);

        }
        System.out.println(qn11.count);

    }

}
```

12. Write a Java program to print a pattern of half diamond using star(*) as shown sample output.

```
*
**
***
****
*****
****
***
**
*
```

Input Format

Number of lines of * to be printed

Output Format

Start pattern as shown in sample output

Sample Input

5

Sample Output

```
*
**
***
****
*****
****
***
**
*
```

```java
import java.util.*;
public class qn12 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int num = in.nextInt();
        for(int i=0 ; i<num;i++){
            for(int j=0;j<i+1;j++){
                System.out.print("*");
            }
            System.out.print("\n");
        }
        for(int i=num;i>0;i--){
            for(int j=1 ;j<i;j++){
                System.out.print("*");
            }
            System.out.print("\n");
        }
    }
}
```

13. Priya has been provided with two unsorted arrays P of size X and Q of size Y of both positive and negative integers. Her work is to identify all pairs from both arrays whose addition is equal to K. Help her finish it.

Input Format

The input contains 3 lines . The first line contains 3 space separated integers X, Y, K. Then in the next two lines are space separated values of the array P and Q respectively.

Output Format

Display the sorted space separated values of all the pairs u,v where u belongs to array P and v belongs to array Q, such that each pair is separated from the other by a ',' without quotes also add a space after the ',' . If no such pair exist display -1.

Constraints

- $1 <= X, Y, K <= 10^6$
- $-10^6 <= P, Q <= 10^6$

Sample Input

5 5 9

1 2 4 5 7

5 6 3 4 8

Sample Output

1 8, 4 5, 5 4

Sample Input

2 2 3

0 2

1 3

Sample Output

0 3, 2 1

Sample Input

87 87 100

22 172 293 175 21 -24 -183 -59 -242 87 -41 -250 153 -25 141 103 274 184 -269 -21 153 -104 -106 180 -242 253 -108 265 63 -252 260 107 -177 -122 -138 25 -187 -38 272 290 298 -294 -186 -75 81 -131 -71 209 -97 -295 -289 20 -268 178 -214 -60 -107 111 -159 94 59 201 88 -94 40 122 266 -72 187 -182 167 202 61 34 226 104 -211 262 -141 16 -201 93 -135 -79 128 -212 236

298 64 -289 -28 -288 -27 80 252 202 20 -183 -4 -41 83 -275 110 71 2 274 3 -185 152 -24 264 263 -159 115 -227 -262 -29 -214 -142 -138 60 -143 75 276 213 -273 -10 -198 196 -242 71 -280 136 7 123 255 248 -80 -71 -300 -252 -104 259 124 -246 -57 1 -46 -290 201 191 75 69 -87 -51 -169 -115 144 -29 -163 187 -246 300 -289 -35 -16 187 182 -116 -106 16 -70 -16 -55

Sample Output

-159 259, -24 124, 20 80, 25 75, 40 60, 93 7, 104 -4, 128 -28, 141 -41, 180 -80, 187 -87, 298 -198

```
import java.util.*;

public class qn13 {
    public static void main(String arg[]){
        Scanner in = new Scanner (System.in);
        int X,Y,K,num=0;
        ArrayList<Integer> a1 = new ArrayList<>();
        ArrayList<Integer> a2 = new ArrayList<>();
        X=in.nextInt();
        Y=in.nextInt();
        K=in.nextInt();
```

```
        for(int i=0 ;i<X;i++){
            num=in.nextInt();
            a1.add(num);
        }
        for(int i=0 ;i<Y;i++){
            num=in.nextInt();
            a2.add(num);
        }
        int count=0;
        for(int a:a1){
            for(int b:a2){
                if(a+b ==K){
                    if(count>0){
                        System.out.print(",");
                        System.out.print(" "+a+" "+b);
                    }
                    else{
                        System.out.print(a+" "+b);
                    }

                    count++;
                }
            }
        }
    }
}
```

14. Given an array **A** consist of **N** number of elements.If the sum of the element is **"even"** print the **sum** of the element.If the sum of the element is **"odd"** print the **product** of the element.

Input Format

The first line of input contains the number of elements **N**

The second line of input represents the elements A_1 , A_2 , A_3 . . . . . A_N

Output Format

Prints the desired result

Sample Input 1:

5

1 2 3 4 5

Sample Output 1:

120

Sample Input 2:

4

Sample Output 2:

10 20 52 51

530400

```java
import java.util.*;
public class qn14 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int n= in.nextInt();
        int p=1 , s=0;
        int num;
        for(int i=0 ;i<n;i++){
            num=in.nextInt();
            p*=num;
            s+=num;
        }

        if((s%2)==0){
            System.out.println(s);
        }
        else{
            System.out.println(p);
        }

    }

}
```

15. A version Managementsystem (VMS) is a repository of files, often the files for the source code of computer programs, with monitored access. Every change made to the source is tracked, along with who made the change, why they made it, and references to problems fixed, or enhancements introduced, by the change.

In this problem we will consider a simplified model of a development project. Let's suppose that there are N source files in the project. All the source files are distinct and numbered from 1 to N.

A VMS which is used for maintaining the project contains two sequences of source files. The first sequence contains M source files that are ignored by the VMS. If a source file is not in the first sequence, then it's considered to be unignored. The second sequence contains K source files that are tracked by the VMS. If a source file is not in the second sequence, then it's considered to be untracked.

A source file can either be or not be in any of these two sequences. Your task is to calculate two values: the number of source files of the project, that are both tracked and ignored, and the number of source files of the project, that are both untracked and unignored.

Input Format

The first line of the input contains three integers N, M and K denoting the number of source files in the project, the number of ignored source files and the number of tracked source files. Assume that the maximum value for N as 50.

The second line contains M distinct integers denoting the sequence A of ignored source files. The sequence is strictly increasing.

The third line contains K distinct integers denoting the sequence B of tracked source files. The sequence is strictly increasing.

Output Format
Output a single line containing two integers: the number of the source files, that are both tracked and ignored, and the number of the source files, that are both untracked and unignored.

Sample 1 Input
7 4 6

1 4 6 7

1 2 3 4 6 7

Sample 1 Output
4 1

Sample 2 Input
4 2 2

1 4

3 4

Sample 2 Output

1 1

```java
import java.util.*;

public class qn15 {
    public static void main(String arg[]){
        int N,M,K;
        Scanner in = new Scanner (System.in);
        N=in.nextInt();
        M=in.nextInt();
        K=in.nextInt();
        ArrayList <Integer> alM = new  ArrayList<Integer>();
        ArrayList <Integer> alK = new  ArrayList<Integer>();
        Set <Integer> set = new HashSet<>();
        int n=0;

        for(int i=0; i<M;i++){
            n=in.nextInt();
            alM.add(n);
            set.add(n);

        }
        for(int i=0; i<K;i++){
            n=in.nextInt();
            alK.add(n);
            set.add(n);

        }
        ArrayList <Integer> al3 = new ArrayList<Integer>(alM);
```

```
        al3.retainAll(alK);
        int num1 , num2;
        num1=al3.size();
        num2=N-set.size();
        System.out.println(num1+" "+num2);

    }

}
```

16. Professor Rakesh is very much interested in Mathematical algorithms. Professor Rakesh Given you two positive integers **'a'** and **'b'**. You are required to calculate the sum of the numbers divisible by 3 and 5, between **'a'** and **'b'** both inclusive and return same.

Input Format

The first and only line of input consists of two integers **a** and **b**

Output Format

Output the sum of numbers that are divisible by 3 and 5

Constraints

1 < **a,b** <1000

Sample Input

12 50

Sample Output

90

```
import java.util.*;
public class qn16 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int a=in.nextInt();
        int b=in.nextInt();
        int sum=0;
        for(int i=12;i<=50;i++){
            if((i%3==0)&&(i%5==0)){
                sum+=i;
            }
        }
        System.out.println(sum);
    }
}
```

17. Mr. Watson likes palindrome numbers very much. So, you should help Mr.watson to find the palindrome numbers between the given limits. You are given a two limits i.e upper limit **'u'** and lower limit **'l'**. You should find the palindrome numbers between the given limits.

Input Format

First and only line of input consists of two integers **u** and **l**

Output Format

Print all the Palindrome numbers between the given limits.

Constraints

$1 < \mathbf{u,l} < 1000$

Sample Input

10 80

Sample Output

11 22 33 44 55 66 77

```java
import java.util.*;
public class qn17 {
    public static boolean check(int a){
        String str= Integer.toString(a);
        String rev="";
        for(int i=str.length()-1;i>=0;i--){
            rev+=str.charAt(i);
        }
        if(str.equals(rev)){
            return true;
        }
        else{
            return false;
        }
    }
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int a,b;
        a=in.nextInt();
        b=in.nextInt();
        for(int i=a ;i<=b;i++){
            if(qn17.check(i)){
                System.out.print(i+" ");
            }
        }


    }
}
```

18. A team is participating in a competition in which one member has to say a string other one has to say a substring from it and the third one has to say the string that has to replace the substring and the last person has to say the modified string. Write a program to obtain a string, substring, and the string that has to replace the substring and prints the modified string.

Input Format
The first line of the input consists of a string.

The second line of the input consists of a substring.

The third line of the input consists of a string that has to replace the substring.

Output Format
Output prints the modified string.

Sample 1 Input
audacious

cio

aaa

Sample 1 Output
audaaaaus

Sample 2 Input
Miss the boat

the

a

Sample 2 Output

Miss a boat

```java
import java.util.*;
public class qn18 {

    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        String str = in.nextLine();
        String sub1=in.nextLine();
        String sub2=in.nextLine();
        str=str.replaceAll(sub1, sub2);
        System.out.println(str);
    }

}
```

19. Vikram hates Vowels. So, he decided to remove all the Vowels in the data set. But, he unable to remove the Vowels. So, help him to remove the Vowels.

Input Format

First line of input consists of an integer **N** representing the number of Strings

Next **N** line consists of strings

Output Format

Print the strings without any Vowels

Constraints

$1 < N < 50$

Sample Input

3

Ajay

kumar

Akash

Sample Output

jy

kmr

ksh

```java
import java.util.*;
public class qn19 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int num = in.nextInt();
        in.nextLine();
        String s;
        String arr[]={"a","e","i","o","u","A","E","I","O","U"};
        ArrayList<ArrayList> al = new ArrayList<>();
        for(int i=0;i<num;i++){
            s=in.nextLine();
            ArrayList<String> st = new ArrayList<>(Arrays.asList(s.split("")));
            for(String e:arr){
                if(st.contains(e)){
                    st.remove(e);
                }
            }
            al.add(st);

        }

        for(ArrayList<String> a:al){
            for(String e:a){
                System.out.print(e);
            }
            System.out.println();
        }
    }
}
```

20. Ramu is in Birla Auditorium. He noticed that the auditorium is decored with an infinite series of electric bulbs indexed from 1. There are 40 switches indexed from 1 to 40 to control those bulbs. Switch with index **N** is connected to the bulbs with indexes that are multiple of **N**. i.e switch 2 is connected to bulb 2,4, 6, 8 .....

So some bulbs are connected to multiple switches and some are not connected to any switch. Ramu started to play with these switches, he wants to know the **K**th glowing bulb from the start if certain switches are in **ON** state. If any of the switch is ON, connected to any bulb then bulb starts glowing. But Ramu has special fond of prime numbers, so he only puts prime indexed switches ON.

Input Format

First line contains 0s and 1s of length 40 that represents the state of electric bulbs (1 is ON and 0 is OFF)

Second line contains one integer K.

Output Format

K^{th} glowing bulb

Constraints

1s are only at prime positions

Sample Input

0110000000000000000000000000000000000000

5

Sample Output

8

Sample Input

0010000000000000000000000000000000000000

5

Sample Output

15

```java
import java.util.*;

public class qn20 {
    public static boolean isPrime(int num){
        boolean fg=true;
        for(int i=2 ; i<num;i++){
            if(num%i ==0){
                fg=false;
                return false;
            }
        }
        if(fg){
            return true;
        }
        else{
            return true;
        }
    }
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        String s=in.nextLine();
```

```
        int i=in.nextInt();
        String [] vt= s.split("");
        ArrayList<Integer> al = new ArrayList<>();
        for(int n=0;n<vt.length;n++){
            if(vt[n].equals("1")){
                if(qn20.isPrime(n+1)){

                    for(int j=1;j<=i;j++){
                        if(!al.contains(j*(n+1))){
                        al.add(j*(n+1));
                    }
                    }
                }
                else{
                    vt[n+1]="0";
                }
            }
        }
        Collections.sort(al);
        System.out.println(al.get(i-1));

    }

}
```

21. Write a Java program to print a character/alphabet pattern as shown in sample output.

F G H I J K
E F G H I
D E F G
C D E
B C
A

Input Format

Number of rows

Output Format

Pattern as shown in sample output

Sample Input

6

Sample Output

F G H I J K
E F G H I
D E F G
C D E
B C
A

```java
import java.util.*;
public class qn21 {
    public static void main(String arg[]){
        String arr[]="ABCDEFGHIJKLMNOPQRSTUVWXYZ".split("");
        Scanner in = new Scanner(System.in);
        int num=in.nextInt();
        for(int i=num;i>=1;i--){
            for(int j=1;j<=i;j++){
                System.out.print(arr[j+i-2]);
            }
            System.out.println();
        }
    }
}
```

22. St. Patrick Convent organizes a project exhibition "Innovative Minds" every year with an objective to provide the platform and unleash the potential of the students by showcasing their innovative projects.
Albert is a science expert and is a topper at his high school. He became interested about the project exhibition and enrolled his name for the same.

Albert's Dad has a cell phone but often seemed to worry about the price plans for his phone that best fits for his usage pattern and monthly expenses.

There are two options, each plan has different costs for daytime minutes, evening minutes and weekend minutes.

Having this as a spark for his project, Albert decided to design a handy application that will input the number of each type of minutes and output the cheapest plan for this usage pattern,using the format shown below. In the case that the two plans are the same price, output both plans.

He needs your help to evaluate his project and suggest corrections.

Hence create a class named BestMobilePlan with the following method.

Create a driver class called Main. In the Main method, obtain input from the user in the console and call the printPlanDetails method present in BestMobilePlan class.

Input Format

Input consists of the usage during daytime,evening and night time separated by a space.

Output Format

Output should print the cheapest plan for this usage pattern. In the case that the two plans are the same price, output both plans.

Sample Input

251 10 60

Sample Output

Plan A costs 51.25

Plan B costs 18.95

Plan B is cheapest

Sample Input

162 61 66

Sample Output

Plan A costs 37.85

Plan B costs 37.85

Plan A and B are the same price

```java
import java.util.*;
class BestMobilePlan{
    int day,eve,night;
    BestMobilePlan(){};
    BestMobilePlan(int a,int b,int c){
        this.day=a;
        this.eve=b;
        this.night=c;
    }
    private double plan_A(){
        double sum=0;
        if(day>100){
            sum+=(day-100)*25;
        }
        sum+=eve*15;
        sum+=night*20;
        return sum/100;
    }
    private double plan_B(){
        double sum=0;
        if(day>250){
            sum+=(day-250)*45;
        }
        sum+=eve*35;
        sum+=night*25;
        return sum/100;
    }

    void printPlanDetails(){
        double A=plan_A();
        double B=plan_B();
        System.out.printf("Plan A costs %.2f \n",A);
        System.out.printf("Plan B costs %.2f \n",B);
        if(A>B){
            System.out.println("Plan B is cheapest");
        }
        else if(A==B){
            System.out.println("Plan A and B are the same price");
        }
        else{
            System.out.println("Plan B is cheapest");
        }
    }

}

class qn22 extends BestMobilePlan{
    qn22(int a,int b,int c){
        super(a,b,c);
    }
```

```
    public static void main (String arg[]){
        int a,b,c;
        Scanner in = new Scanner(System.in);
        a=in.nextInt();
        b=in.nextInt();
        c=in.nextInt();

        qn22 obj = new qn22(a,b,c);
        obj.printPlanDetails();
    }
}
```

23. Little monk loves good string. Good String is a string that only contains vowels(a , e, i, o, u). Now, his teacher gave him a string S. Little monk is wondering what is the length of the longest good string which is a substring of S.

Note: Strings contains only lower case English Alphabets

Input Format

First line contains a string S, where S denotes the length of the string.

Output Format

Print an integer denoting the length of the longest good substring, that is substring consists of only vowels.

Sample Input

Sample Output

abcaac

2

Sample Input:

mzbmweyydiadtlcouegmdbyfwurpwbpuvhifnuapwyndmhtqvkgkbhtytszotwflegsjzzszfwtzfpnscg
uemwrczqxycivdqnkypnxnnpmuduhznoaquudhavmcwfwujpcmiggjmcmkkbnjfeodxkgjgwxtrxin
giqquhuwqhdswxxrxuzzfhkplwunfagppcoildagktgdarveusjuqfistulgbglwmfgzrnyxryetwzhlnfewc
zmnoozlqatugmdjwgzcfabbkoxyjxkatjmpprswkdkobdagwdwxsufeesrvncbszcepigpbzuzoootorzfs
kcwbqorvwdrmklfdczatfarqdkelalxzxillkfdvpfpxabqlngdscrentzamztvvcvrtcmbqlizijdwtuyfrxols
ysxlfebpolcmqsppmrfkyunydtmwbexsngxhwvroandfqjamzkpttslildlrkjoyrpxugiceahgiakevsjoad
mkfnkswrawkjxwcmcciabzbrskzazjqtlkiqydptpk

Sample Output:

3

```
import java.util.*;
public class qn23 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        String s= in.nextLine();
        char arr[]=s.toCharArray();
        List <String> vov = new ArrayList<>(Arrays.asList("aeiouAEIOU".split("")));
        String temp;
        String Max_String = "" ;
        int max_int=0;
        for(char a: arr){
            String e=Character.toString(a);
            if(vov.contains(e)){
```

```
                Max_String+=e;
            }
            else{
                if(max_int<Max_String.length()){
                    max_int=Max_String.length();
                }
                Max_String="";
            }
        }
        System.out.println(max_int);
    }
}
```

24.Ram gifted a new Vespa scooter to Jannu. But Jannu's father instructed Jannu to drive within speed limits. So every day Jannu's father asks the distance traveled(in km) and time taken(in hrs) to cover the distance. He calculates the speed using given values and check Jannu's speed.

Help Jannu's father to calculate the speed.

Note : Display the result in km/hr unit

Input Format

Distance traveled in first line

Time taken to travel in second line

Output Format

Speed with units as shown in sample output

Sample Input

200

10

Sample Output

20.00 km/hr

Sample Input

50.5

10

Sample Output

5.05 km/hr

Sample Input

90

4.5

Sample Output

20.00 km/hr

```java
import java.util.*;
class qn24
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        float a = sc.nextFloat();
        float b = sc.nextFloat();
        float c;
        c = a/b;
        System.out.println(c +" Km/hr");
    }
}
```

25. Priya has been provided with two unsorted arrays P of size X and Q of size Y of both positive and negative integers. Her work is to identify all pairs from both arrays whose addition is equal to K. Help her finish it.

Input Format

The input contains 3 lines . The first line contains 3 space separated integers X, Y, K. Then in the next two lines are space separated values of the array P and Q respectively.

Output Format

Display the sorted space separated values of all the pairs u,v where u belongs to array P and v belongs to array Q, such that each pair is separated from the other by a ',' without quotes also add a space after the ',' . If no such pair exist display -1.

Constraints

- **$1 <= X, Y, K <= 10^6$**
- **$-10^6 <= P, Q <= 10^6$**

Sample Input

5 5 9

1 2 4 5 7

5 6 3 4 8

Sample Output

1 8, 4 5, 5 4

Sample Input

2 2 3

0 2

1 3

Sample Output

0 3, 2 1

Sample Input

87 87 100

22 172 293 175 21 -24 -183 -59 -242 87 -41 -250 153 -25 141 103 274 184 -269 -21 153 -104 -106 180 -242 253 -108 265 63 -252 260 107 -177 -122 -138 25 -187 -38 272 290 298 -294 -186 -75 81 -131 -71 209 -97 -295 -289 20 -268 178 -214 -60 -107 111 -159 94 59 201 88 -94 40 122 266 -72 187 -182 167 202 61 34 226 104 -211 262 -141 16 -201 93 -135 -79 128 -212 236

298 64 -289 -28 -288 -27 80 252 202 20 -183 -4 -41 83 -275 110 71 2 274 3 -185 152 -24 264 263 -159 115 -227 -262 -29 -214 -142 -138 60 -143 75 276 213 -273 -10 -198 196 -242 71 -280 136 7 123 255 248 -80 -71 -300 -252 -104 259 124 -246 -57 1 -46 -290 201 191 75 69 -87 -51 -169 -115 144 -29 -163 187 -246 300 -289 -35 -16 187 182 -116 -106 16 -70 -16 -55

Sample Output

-159 259, -24 124, 20 80, 25 75, 40 60, 93 7, 104 -4, 128 -28, 141 -41, 180 -80, 187 -87, 298 -198

```java
import java.util.*;

public class qn25 {
    public static void main(String arg[]){
        Scanner in = new Scanner (System.in);
        int X,Y,K,num=0;
        ArrayList<Integer> a1 = new ArrayList<>();
        ArrayList<Integer> a2 = new ArrayList<>();
        X=in.nextInt();
        Y=in.nextInt();
        K=in.nextInt();
        for(int i=0 ;i<X;i++){
            num=in.nextInt();
            a1.add(num);
        }
        for(int i=0 ;i<Y;i++){
            num=in.nextInt();
            a2.add(num);
        }
        int count=0;
        for(int a:a1){
            for(int b:a2){
                if(a+b ==K){
                    if(count>0){
                        System.out.print(",");
                        System.out.print(" "+a+" "+b);
                    }
                    else{
                        System.out.print(a+" "+b);
```

```
                    }

                    count++;
                }
            }
        }
    }
}
```

26. Peter is working in the Data analyst company. His team leader gave a task to Peter. The task is, peter wants to find the number of occurrence of a certain letter in the given string. Help him to complete this task.

Input Format

First line of Input contains the String **S**

Next line contains the Character **C**

Output Format

Output the number of occurrences of the given letter.

Constraints

String should contain only alphabets

Sample Input

madam

a

Sample Output

2

```
import java.util.*;
public class qn26 {
    public static void main(String arg[]){
        String input ;
        Scanner in = new Scanner (System.in);
        input=in.nextLine();
        Map<String,Integer> map = new HashMap<>();
        ArrayList<String> newarray = new ArrayList<>(Arrays.asList(input.split("")));
        for(String s:newarray){
            if(map.containsKey(s)){
                int num=map.get(s);
                map.replace(s, num, num+1);
            }
            else{
                map.put(s, 1);
            }
        }

        String key ;
        key = in.nextLine();
```

```
        System.out.println(map.get(key));

    }

}
```

27.Create a driver class called Main. In the Main method, obtain input from the user in the console and display the maximum size of the stack of cards which can be exchanged between the two players by calling the findValue method present in the Cards class.

Input Format
The input consists of the amount of collectable cards that gary have to change and dory have to change separated by a space.

Output Format
Output the maximum size of the stack of cards which can be exchanged between the two players, in a single line. The Warners Club arranged for a Recreation Fair for kids and adults for the winter season ahead. The Club offered outdoor activities for adults like hiking, cycling, power soccer and indoor activities like chess, carom and card games for kids below 10 years. Gary and Dory accompanied their parents for the fair and wanted to play the cricket collectable card game as they are crazy about the collectable cards.

In their spare time, they usually have a way of playing some game involving such cards. Both also have the habit of exchanging the repeated cards with their friends. That day at the Fair, Gary and Dory thought about a different game. With the cards in hand, each person tried to make an exchange with the other person following this simple rule: each person must count how many cards he owned. After this, they had to split these cards into stacks, all of it with the same size, as large as it was possible for both players. Then, each one chooses one of the friend's card stacks to receive.

For example, if Gary and Dory would change the cards with respectively 8 and 12 cards each, both must put his cards in stacks of four cards (Gary would have two stacks and Dory had three stacks), and both choose a stack from his friend to receive it.

Help Gary and Dory find the maximum size of the stack of cards which can be exchanged between the two players.

Hence create a class named Cards with the following method.

Sample 1 Input

8 12

Sample 1 Output
4

Sample 2 Input

9 27

Sample 2 Output

9

```java
import java.util.*;

class qn27
{
    public static void main(String[] args)
    {
        int c;
        Scanner sc = new Scanner(System.in);
        int a=sc.nextInt() ;
        int b=sc.nextInt();
        c=(a>b)?b:a;
        for(int i=c;i>1;i--){
            if(a%i==0 && b%i==0){
                System.out.println(i);
                break;
            }
        }
    }
}
```

28.. Given an array **A** consist of **N** number of elements.If the sum of the element is **"even"** print the **sum** of the element.If the sum of the element is **"odd"** print the **product** of the element.

Input Format

The first line of input contains the number of elements **N**

The second line of input represents the elements A_1 , A_2 , A_3 . . . . . A_N

Output Format

Prints the desired result

Sample Input 1:

5

1 2 3 4 5

Sample Output 1:

120

Sample Input 2:

4

Sample Output 2:

10 20 52 51

530400

```java
import java.util.*;

public class qn28 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int n= in.nextInt();
        int p=1 , s=0;
        int num;
        for(int i=0 ;i<n;i++){
            num=in.nextInt();
            p*=num;
            s+=num;
```

```
        }

        if((s%2)==0){
            System.out.println(s);
        }
        else{
            System.out.println(p);
        }

    }

}
```

29. Ria wants to gift a bouquet to her father on his birthday and asked for help from her mother Jenni. Jenni gives **N** flower sticks numbered 1 to **N** to Ria and tells her to arrange it in the bouquet in a particular order. She asks her to arrange the first **K** flower sticks in the order of their increasing length and the remaining sticks in an order of their decreasing length.

Write an Java Program to find the final arrangement of the flower sticks in which Ria gifted the bouquet to her father.

Input Format

First line of input contains **N** and **K**

Next line contains **N** space separated integers

Output Format

Return a list of integers representing the final pattern of the flower sticks in which Ria gifted the bouquet to her father

Constraints

K < N

Sample Input

8 4

3 4 6 7 2 9 8 1

Sample Output

1 2 3 4 9 8 7 6

```
import java.util.*;
public class qn29 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int N = in.nextInt();
        int K = in.nextInt();
        int e;
        ArrayList<Integer> al = new ArrayList<>();
        for( int i=0 ;i<N;i++){
            e=in.nextInt();
            al.add(e);
        }
```

```
        Collections.sort(al);
        for(int i=0 ;i<K;i++){
            System.out.print(al.get(0)+" ");
            al.remove(0);
        }

        for(int i=al.size()-1;i>=0;i--){
            System.out.print(al.get(i)+" ");
        }
    }
}
```

30. Mike is typewriter. His boss, assigned him a new task. Mike should reverse the each words in the string **S** . He can reverse a single word, but unable to reverse the multiple words in the string. So, help him to complete this task.

Input Format

First and only line of input contains the string.

Output Format

Print the reversed string

Constraints

String should contain alphabets and white spaces.

Sample Input

varun is a teacher

Sample Output

nurav si a rehcaet

ii. Harish given you a number **N** and **K**. You should create a string **P** by concatenating the **N** **K** times. For example, **N**=12 ,**K**=3, then P=121212. Your task is to find the excellent digit of the **P**. Excellent digit of the number has following rules:

- If **P** has only digit, then its excellent digit is **P** .
- Otherwise, the excellent digit of **P** is equal to the excellent digit of the sum of the digits of **P**.

Input Format

First line of input contains two space separated integers **N** and **K**.

Output Format

Output the excellent digit.

Constraints

1 <= **N** <= 100

1 <= **K** <= 100

Sample Input

148 3

Sample Output

```java
import java.util.*;
public class qn30_1 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        String str = in.nextLine();
        String arr[] = str.split(" ");
        String NewStr="";
        for(int i=0 ; i<arr.length;i++){
            String s= arr[i];
            StringBuilder stb = new StringBuilder();
            stb.append(s);
            s=stb.reverse().toString();
            NewStr=NewStr+s+" ";
        }
        System.out.println(NewStr);
    }
}
```

31. Ahmed is a English professor in XYZ college of technology. The college management assigned him a new task. He has to convert a string to camel case i.e capitalise the first letter of each word. Help him to convert the string to camel case .

Input Format

First and only line of input contains a string.

Output Format

Print the string in camel case

Constraints

The First letter of each word in the string should contain only alphabets

Sample Input

rohit is a cricket player

Sample Output

Rohit Is A Cricket Player

Sample Input

develop a code to print the binary value of the input decimal number. Input should be accepted from the command line.

Sample Output

Develop A Code To Print The Binary Value Of The Input Decimal Number. Input Should Be Accepted From The Command Line.

```java
import java.util.*;

public class qn31 {

    public static void main(String arg[]){

        Scanner in = new Scanner (System.in);
        String str = in.nextLine();
        String arr[] = str.split(" ");
        String temp="";
        for(int i=0 ;i<arr.length;i++){
            temp+=Character.toString(arr[i].charAt(0)).toUpperCase();
            temp+=arr[i].substring(1, arr[i].length());
            temp+=" ";
        }
```

```
        System.out.println(temp);
    }

}
```

32. Fathima has a numeric string. She has to split the numeric string into two or more integers, such that

1. Difference between current and previous number is 1.

2. No number contains leading zeroes

If it is possible to separate a given numeric string then print "Possible" followed by the first number of the increasing sequence, else print "Not Possible"

For Ex:

Case 1 :

If the input is 7980

Output: Possible 79

String can be splitted into 79 and 80 and the difference is 1

Case 2:

If the input is 124

Output: Not Possible

String can be splitted into '1','2' and '4". The difference between 1 and 2 is 1, but 2 and 4 is 2. So it can't satisfy the condition. The output will be "Not Possible"

Input Format

Input the String(Which contains only numbers. Alphabets and Special characters are not allowed)

Output Format

Display "Possible" followed by the first number of the increasing sequence. If not display "Not Possible"

Constraints

0 <= length(numeric string) <= 35

Sample Input

7980

Sample Output

Possible 79

```
import java.io.*;
import java.util.*;

class qn32 {

    public static void split(String str)
    {
        int len = str.length();

        if (len == 1) {
            System.out.println("Not Possible");
            return;
        }
```

```
        String s1 = "", s2 = "";
        long num1, num2;

        for (int i = 0; i <= len / 2; i++) {

            int flag = 0;

            s1 = str.substring(0, i + 1);
            num1 = Long.parseLong((s1));
            num2 = num1 + 1;

            s2 = Long.toString(num2);
            int k = i + 1;
            while (flag == 0) {
                int l = s2.length();
                if (k + l > len) {
                    flag = 1;
                    break;
                }

                if ((str.substring(k, k + l).equals(s2))) {
                    flag = 0;

                    num2++;
                    k = k + l;
                    if (k == len)
                        break;
                    s2 = Long.toString(num2);
                    l = s2.length();
                    if (k + 1 > len) {

                        flag = 1;
                        break;
                    }
                }

                else
                    flag = 1;
            }

            if (flag == 0) {
                System.out.println("Possible"
                                   + " " + s1);
                break;
            }
             else if (flag == 1 && i > len / 2 - 1) {
                System.out.println("Not Possible");
                break;
            }
```

```
        }
    }
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        String str = in.nextLine();
        split(str);
    }
}
```

33.A parking lot charges Rs.30 as a minimum fee to park a vehicle for up to 3 hours. An additional charge of Rs.5.00 per hour will be added if it exceeds three hours. for 24 hours the parking fees are Rs.80.00.Write a program to define an array and read the vehicle registration number, and hours parked for each customer and calculate the parking charges for 'n' customers and display the output.

Input Format
The first line of the output consists of the number of customers.

The next n lines consists of the vehicle registration number and total parking hours.

Output Format
The output consists of the vehicle number, hours parked, and charges separated by a space.

Sample 1 Input
3

1867 3

5382 5

2407 24

Sample 1 Output
1867 3 30.00

5382 5 40.00

2407 24 80.00

```
import java.util.*;
public class qn33 {
    public static double check(int i){
        double sum=0;
        if(i==24){
            sum=80;
            return sum;
        }
        if(i<=3){
            sum+=30;
            return (double)sum;
        }
        else{
            sum=30+((i-3)*5);
            return sum;

        }

    }
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
```

```
        int n = in.nextInt();
        Map<Integer,Integer> mp=new HashMap<>();
        ArrayList<Integer> al = new ArrayList<>();
        for(int i=0;i<n;i++){
            int no , hr;
            no=in.nextInt();
            hr=in.nextInt();
            mp.put(no, hr);
            al.add(no);

        }
        for(int i:al){
            double rs = qn33.check(mp.get(i));
            System.out.println(i+" "+mp.get(i)+" "+rs);
        }

    }

}
```

34. Ramu is in Birla Auditorium. He noticed that the auditorium is decored with an infinite series of electric bulbs indexed from 1. There are 40 switches indexed from 1 to 40 to control those bulbs. Switch with index **N** is connected to the bulbs with indexes that are multiple of **N**. i.e switch 2 is connected to bulb 2,4, 6, 8 .....

So some bulbs are connected to multiple switches and some are not connected to any switch. Ramu started to play with these switches, he wants to know the **K**th glowing bulb from the start if certain switches are in **ON** state. If any of the switch is ON, connected to any bulb then bulb starts glowing. But Ramu has special fond of prime numbers, so he only puts prime indexed switches ON.

Input Format
First line contains 0s and 1s of length 40 that represents the state of electric bulbs (1 is ON and 0 is OFF)

Second line contains one integer K.

Output Format
K^{th} glowing bulb

Constraints
1s are only at prime positions

Sample 1 Input
0110000000000000000000000000000000000000

5

Sample 1 Output
8

Sample 2 Input
0010000000000000000000000000000000000000

5

Sample 2 Output
15

```
import java.util.*;
```

```java
public class qn34 {
    public static boolean isPrime(int num){
        boolean fg=true;
        for(int i=2 ; i<num;i++){
            if(num%i ==0){
                fg=false;
                return false;
            }
        }
        if(fg){
            return true;
        }
        else{
            return true;
        }
    }
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        String s=in.nextLine();
        int i=in.nextInt();
        String [] vt= s.split("");
        ArrayList<Integer> al = new ArrayList<>();
        for(int n=0;n<vt.length;n++){
            if(vt[n].equals("1")){
                if(qn20.isPrime(n+1)){

                    for(int j=1;j<=i;j++){
                        if(!al.contains(j*(n+1))){
                        al.add(j*(n+1));
                    }
                    }
                }
                else{
                    vt[n+1]="0";
                }
            }
        }
        Collections.sort(al);
        System.out.println(al.get(i-1));

    }

}
```

35. Ahmed is a English professor in XYZ college of technology. The college management assigned him a new task. He has to convert a string to camel case i.e capitalise the first letter of each word. Help him to convert the string to camel case .

Input Format

First and only line of input contains a string.

Output Format

Print the string in camel case

Constraints

The First letter of each word in the string should contain only alphabets

Sample Input

rohit is a cricket player

Sample Output

Rohit Is A Cricket Player

Sample Input

develop a code to print the binary value of the input decimal number. Input should be accepted from the command line.

Sample Output

Develop A Code To Print The Binary Value Of The Input Decimal Number. Input Should Be Accepted From The Command Line.

```java
import java.util.*;
public class qn35 {
    public static void main(String arg[]){
        Scanner in = new Scanner (System.in);
        String str = in.nextLine();
        String arr[] = str.split(" ");
        String temp="";
        for(int i=0 ;i<arr.length;i++){
            temp+=Character.toString(arr[i].charAt(0)).toUpperCase();
            temp+=arr[i].substring(1, arr[i].length());
            temp+=" ";
        }
        System.out.println(temp);
    }

}
```

36. Louis was celebrating his 10th Birthday and his parents wished to make his birthday more special by throwing a surprise bash, inviting all their friends, relatives and neighbors. The little mathematics geek Louis has another surprise waiting for him when he had to cut his favorite Choco vanilla cake. The cake is a rectangular cake and it consists of $m \times n$ ($1 \leq m, n \leq 1000$) squares. His friends now called him out for a challenge.

The challenge is that, Louis has to break the cake up into 1×1 pieces (individual squares) and find what is the minimum number of times that he breaks the choco vanilla cake, or pieces thereof, in order to achieve this?

Note that he cannot stack pieces of the cake and break them, because the choco vanilla cake is thick. As an example, a 2×2 cake requires 3 breaks. First he can break it in half, then break each of the halves in half. He cannot break it in half, stack the two 1×2 pieces, and then use only one more break to achieve his goal.

Input Format
First line of the input consists of an integer, the dimensions m of the choco vanilla cake.

Second line of the input consists of an integer, the dimensions n of the choco vanilla cake.

Output Format
Output the minimum number of times that he breaks the choco vanilla cake

Refer sample input and output for formatting specifications.

Sample 1 Input
1

2

Sample 1 Output
Minimum number of times is 1

Sample 2 Input
2

2

Sample 2 Output
Minimum number of times is 3

```java
import java.util.*;
public class qn36 {
    public static void main(String arg[]){
        Scanner in =  new Scanner(System.in);
        int m=in.nextInt();
        int n= in.nextInt();
        System.out.println("Minimum number od times is "+(m*n-1));
    }

}
```

37. Write a Java program to delete an element from the array.

Input Format

The first line of the input consists of the number of elements.

Second input is the array elements separated by a space.

The third input is the element to be deleted.

Output Format

The output prints the array after the deletion of the specified element.

Constraints

N > 0

Sample Input

5

2 4 3 7 8

7

Sample Output

2 4 3 8

```java
import java.util.*;
public class qn37 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int n= in.nextInt();
        ArrayList<Integer> al = new ArrayList<>();
        int e=0;
        for (int i=0;i<n;i++){
            e=in.nextInt();
            al.add(e);
        }
        int num = in.nextInt();
        al.remove((Integer)num);
        for(int i:al){
            System.out.print(i+" ");
        }
    }
}
```

38.. St. Patrick Convent organizes a project exhibition "Innovative Minds" every year with an objective to provide the platform and unleash the potential of the students by showcasing their innovative projects.

Albert is a science expert and is a topper at his high school. He became interested about the project exhibition and enrolled his name for the same.

Albert's Dad has a cell phone but often seemed to worry about the price plans for his phone that best fits for his usage pattern and monthly expenses.

There are two options, each plan has different costs for daytime minutes, evening minutes and weekend minutes

Having this as a spark for his project, Albert decided to design a handy application that will input the number of each type of minutes and output the cheapest plan for this usage pattern, using the format shown below. In the case that the two plans are the same price, output both plans.

He needs your help to evaluate his project and suggest corrections.

Hence create a class named BestMobilePlan with the following method.

Create a driver class called Main. In the Main method, obtain input from the user in the console and call the printPlanDetails method present in BestMobilePlan class.

Input Format

Input consists of the usage during daytime,evening and night time separated by a space.

Output Format

Output should print the cheapest plan for this usage pattern. In the case that the two plans are the same price, output both plans.

Sample Input

251 10 60

Sample Output

Plan A costs 51.25

Plan B costs 18.95

Plan B is cheapest

Sample Input

162 61 66

Sample Output

Plan A costs 37.85

Plan B costs 37.85

Plan A and B are the same price

```
import java.util.*;
class BestMobilePlan{
    int day,eve,night;
    BestMobilePlan(){};
    BestMobilePlan(int a,int b,int c){
        this.day=a;
        this.eve=b;
        this.night=c;
    }
    private double plan_A(){
        double sum=0;
        if(day>100){
            sum+=(day-100)*25;
        }
        sum+=eve*15;
        sum+=night*20;
        return sum/100;
    }
    private double plan_B(){
        double sum=0;
        if(day>250){
            sum+=(day-250)*45;
        }
        sum+=eve*35;
        sum+=night*25;
        return sum/100;
    }

    void printPlanDetails(){
```

```
        double A=plan_A();
        double B=plan_B();
        System.out.printf("Plan A costs %.2f \n",A);
        System.out.printf("Plan B costs %.2f \n",B);
        if(A>B){
            System.out.println("Plan B is cheapest");
        }
        else if(A==B){
            System.out.println("Plan A and B are the same price");
        }
        else{
            System.out.println("Plan B is cheapest");
        }
    }

}

class qn38 extends BestMobilePlan{
    qn38(int a,int b,int c){
        super(a,b,c);
    }
    public static void main (String arg[]){
        int a,b,c;
        Scanner in = new Scanner(System.in);
        a=in.nextInt();
        b=in.nextInt();
        c=in.nextInt();

        qn38 obj = new qn38(a,b,c);
        obj.printPlanDetails();
    }
}
```

39. The much awaited event at the entertainment industry every year is the "Screen Awards". This year the event is going to be organized on December 25 to honour the Artists for their professional excellence in Cinema. The Organizers has this time decided to launch an online portal to facilitate easy booking of the Award show's tickets.

They specifically wanted to provide an option for bulk booking in the portal, wherein there are many discounts announced. Write a program to help the Organizers to create the portal as per the requirement given below.

Given the ticket cost as 'X'.

If the number of tickets purchased is less than 50, there is no discount.

If the number of tickets purchased is between 50 and 100 (both inclusive), then 10% discount is offered.

If the number of tickets purchased is between 101 and 200(both inclusive), 20% discount is offered.

If the number of tickets purchased is between 201 and 400(both inclusive), 30% discount is offered.

If the number of tickets purchased is between 401 and 500(both inclusive), 40% discount is offered.

If the number of tickets purchased is greater than 500, then 50% discount is offered.

Input Format
First line of the input is an integer that corresponds to the cost of the ticket 'X'.

Second line of the input is an integer that corresponds to the number of tickets purchased.

Output Format
Output should display a double value, which gives the total expenses in purchasing the tickets after discounts. Display the output correct to 2 decimal places.

Sample 1 Input
100

5

Sample 1 Output
500.00

Sample 2 Input
100

300

Sample 2 Output
21000.00

```java
import java.util.*;
public class qn39 {
    public static void main (String arg[]){
        Scanner in = new Scanner(System.in);
        int X= in.nextInt();
        int num = in.nextInt();
        if(num<50){
            System.out.printf("%.2f",(double)X*num);
        }
        else if(num>=50 && num<=100){
            double rs=X*num;
            rs=rs-(rs*10/100);
            System.out.printf("%.2f",rs);

        }
        else if(num>=101 && num<=200){
            double rs=X*num;
            rs=rs-(rs*20/100);
            System.out.printf("%.2f",rs);

        }
        else if(num>=201 && num<=400){
            double rs=X*num;
            rs=rs-(rs*30/100);
            System.out.printf("%.2f",rs);

        }
    }
}
```

40. Hari is a civil engineer who is designing a fountain in square shape with water sprinklers in the edges with n number of steps. He needs to draw a sketch of the fountain in top view with the step number at the edges of the square.

Write a Java program to help him in printing the pattern with n number of steps.

Input Format
Input is an positive integer describing the step levels of the fountain.

Output Format
Output consists of the pattern of numbers as described in the question for n number of step levels.

Sample 1 Input
5

Sample 1 Output

```
1         1
 2       2
  3     3
   4   4


    5
   4 4
  3   3
 2     2
1       1
```

```java
import java.util.*;
public class qn40 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int num=in.nextInt();
        for(int i=1;i<=num;i++){
            for(int a=1;a<=num;a++){
                if(a==i){
                    System.out.print(a);
                }
                else{
                    System.out.print(" ");
                }
            }
            for(int a=num-1;a>=1;a--){
                if(a==i){
                    System.out.print(a);
                }
                else{
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
        for(int i=num-1;i>=1;i--){
            for(int a=1;a<=num;a++){
```

```
                if(a==i){
                    System.out.print(a);
                }
                else{
                    System.out.print(" ");
                }
            }
            for(int a=num-1;a>=1;a--){
                if(a==i){
                    System.out.print(a);
                }
                else{
                    System.out.print(" ");
                }
            }
            System.out.println();
        }
    }
}
```

41. Write a Java program to display the multiples of a number '**n**' upto the given limit '**L**' and print the sum of all the multiples in it.

Input Format

First and only line of input contains two integers **n** and **L**

Output Format

First **L** lines contains the multiples of the given number **n**

Next line consists of sum of the multiples.

Constraints

$2 < \mathbf{n},\mathbf{L} < 1000$

Sample Input

12 10

Sample Output

12
24
36
48
60
72
84
96
108
120
660

```
import java.util.*;
public class qn41 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
```

```
        int n=in.nextInt();
        int m=in.nextInt();
        int sum=0;
        for(int i=1;i<=m;i++){
            System.out.println(n*i);
            sum+=n*i;
        }
        System.out.println(sum);
    }
}
```

42. Kishore given you a number **N** as a string with no leading zeros. Your task is to determine the sum of all integer values of substrings of the string. For example, if the string is 42 , the substrings are 4,2 and 42 . Their sum is 48.

Given an integer as a string, sum all of its substrings cast as integers. As the number may become large, return the value modulo 10^9+7.

Input Format

First line of input contains an integer as a string without leading zeros.

Output Format

Output the sum of substrings

Constraints

$1 <= n <= 2*10^5$

Sample Input

16

Sample Output

23

```
import java.util.*;
public class qn42 {
    public static void main(String arg[]){
        Scanner in= new Scanner(System.in);
        String str=in.nextLine();
        ArrayList<String> al = new ArrayList<>();
        for(int j=0;j<str.length();j++){
            for(int k=0;k<str.length();k++){
                String s="";
                for(int i=j;i<=k;i++){
                    s+=str.charAt(i);
                }
                if(!s.equals("")){
                    al.add(s);
                }
            }
        }
        int sum=0;
        for(String s:al){
            sum+=Integer.parseInt(s);
```

```
        }
        System.out.println(sum);
    }
}
```

43. A team is participating in a competition in which one member has to say a string other one has to say a substring from it and the third one has to say the string that has to replace the substring and the last person has to say the modified string. Write a program to obtain a string, substring, and the string that has to replace the substring and prints the modified string.

Input Format
The first line of the input consists of a string.

The second line of the input consists of a substring.

The third line of the input consists of a string that has to replace the substring.

Output Format
Output prints the modified string.

Sample 1 Input
audacious

cio

aaa

Sample 1 Output
audaaaaus

Sample 2 Input
Miss the boat

the

a

Sample 2 Output
Miss a boat

```
import java.util.*;
public class qn43 {

    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        String str = in.nextLine();
        String sub1=in.nextLine();
        String sub2=in.nextLine();
        str=str.replaceAll(sub1, sub2);
        System.out.println(str);
    }

}
```

44. Write a Java program to find smallest and largest word in the given string.

Input Format

The input consists of the sentence.

Output Format

The output prints the smallest and largest word in the sentence.

Sample Input

an apple for a day

Sample Output

Smallest word: a

Largest word: apple

```
import java.util.*;
public class qn44 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        String str = in.nextLine();
        String smallest=str ,largest="";
        String arr[]=str.split(" ");
        for(String e:arr){
            if(smallest.length()>e.length()){
                smallest=e;
            }
            if(largest.length()<e.length()){
                largest=e;
            }

        }
        System.out.println("Smallest word: "+smallest);
        System.out.println("Largest word: "+largest);
    }
}
```

45. St. Patrick Convent organizes a project exhibition "Innovative Minds" every year with an objective to provide the platform and unleash the potential of the students by showcasing their innovative projects.

Albert is a science expert and is a topper at his high school. He became interested about the project exhibition and enrolled his name for the same.

Albert's Dad has a cell phone but often seemed to worry about the price plans for his phone that best fits for his usage pattern and monthly expenses.

There are two options, each plan has different costs for daytime minutes, evening minutes and weekend minutes.

Having this as a spark for his project, Albert decided to design a handy application that will input the number of each type of minutes and output the cheapest plan for this usage pattern, using the format shown below. In the case that the two plans are the same price, output both plans.

He needs your help to evaluate his project and suggest corrections.

Hence create a class named BestMobilePlan with the following method.

Create a driver class called Main. In the Main method, obtain input from the user in the console and call the printPlanDetails method present in BestMobilePlan class.

Input Format

Input consists of the usage during daytime,evening and night time separated by a space.

Output Format

Output should print the cheapest plan for this usage pattern. In the case that the two plans are the same price, output both plans.

Sample Input

251 10 60

Sample Output

Plan A costs 51.25

Plan B costs 18.95

Plan B is cheapest

Sample Input

162 61 66

Sample Output

Plan A costs 37.85

Plan B costs 37.85

Plan A and B are the same price

```
import java.util.*;
class BestMobilePlan{
    int day,eve,night;
    BestMobilePlan(){};
    BestMobilePlan(int a,int b,int c){
        this.day=a;
        this.eve=b;
        this.night=c;
    }
    private double plan_A(){
        double sum=0;
        if(day>100){
            sum+=(day-100)*25;
        }
        sum+=eve*15;
        sum+=night*20;
        return sum/100;
    }
    private double plan_B(){
```

```
        double sum=0;
        if(day>250){
            sum+=(day-250)*45;
        }
        sum+=eve*35;
        sum+=night*25;
        return sum/100;
    }

    void printPlanDetails(){
        double A=plan_A();
        double B=plan_B();
        System.out.printf("Plan A costs %.2f \n",A);
        System.out.printf("Plan B costs %.2f \n",B);
        if(A>B){
            System.out.println("Plan B is cheapest");
        }
        else if(A==B){
            System.out.println("Plan A and B are the same price");
        }
        else{
            System.out.println("Plan B is cheapest");
        }
    }

}

class qn45 extends BestMobilePlan{
    qn45(int a,int b,int c){
        super(a,b,c);
    }
    public static void main (String arg[]){
        int a,b,c;
        Scanner in = new Scanner(System.in);
        a=in.nextInt();
        b=in.nextInt();
        c=in.nextInt();

        qn45 obj = new qn45(a,b,c);
        obj.printPlanDetails();
    }
}
```

46. Akash is very much interested in sorting the arrays. One day, he tries to sort the array. But , there is a condition that he should sort the array according to the product of the digits. Help him to sort the array.

Input Format

First line of input contains the integer **N** denotes size of the array **arr**.

Next line contains **N** space separated integers.

Output Format

Output the sorted array

Constraints

2 <= **N** <= 100

Sample Input

5

12 10 102 31 15

Sample Output

10 102 12 31 15

```java
import java.util.*;
public class qn46 {
    public static int prod(int num){
        int p =1;
        while(num!=0){
            p*=num%10;
            num/=10;
        }
        return p;
    }
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int num = in.nextInt();
        int e;
        List<Integer> al = new ArrayList<>();
        for(int i=0;i<num;i++){
            e=in.nextInt();
            al.add(e);
        }
        ArrayList<Integer> sorted = new ArrayList<>();
        int size=al.size();
        for(int i=0 ;i<size;i++){
            int min=1000;
            int elm=0;
            for(int a:al){
                if(qn46.prod(a)<min){
                    min=qn46.prod(a);
                    elm=a;
                }
            }
            sorted.add(elm);
            al.remove((Integer)elm);
        }
```

```
        for(int i:sorted){
            System.out.print(i+" ");
        }
    }
}
```

47. The Warners Club arranged for a Recreation Fair for kids and adults for the winter season ahead. The Club offered outdoor activities for adults like hiking, cycling, power soccer and indoor activities like chess, carom and card games for kids below 10 years. Gary and Dory accompanied their parents for the fair and wanted to play the cricket collectable card game as they are crazy about the collectable cards.

In their spare time, they usually have a way of playing some game involving such cards. Both also have the habit of exchanging the repeated cards with their friends. That day at the Fair, Gary and Dory thought about a different game. With the cards in hand, each person tried to make an exchange with the other person following this simple rule: each person must count how many cards he owned. After this, they had to split these cards into stacks, all of it with the same size, as large as it was possible for both players. Then, each one chooses one of the friend's card stacks to receive.

For example, if Gary and Dory would change the cards with respectively 8 and 12 cards each, both must put his cards in stacks of four cards (Gary would have two stacks and Dory had three stacks), and both choose a stack from his friend to receive it.

Help Gary and Dory find the maximum size of the stack of cards which can be exchanged between the two players.

Hence create a class named Cards with the following method.

Create a driver class called Main. In the Main method, obtain input from the user in the console and display the maximum size of the stack of cards which can be exchanged between the two players by calling the findValue method present in the Cards class.

Input Format
The input consists of the amount of collectable cards that gary have to change and dory have to change separated by a space.

Output Format
Output the maximum size of the stack of cards which can be exchanged between the two players, in a single line.

Sample 1 Input
8 12

Sample 1 Output
4

Sample 2 Input
9 27

Sample 2 Output
9

```
import java.util.*;
class qn47
{
    public static void main(String[] args)
    {
        int c;
        Scanner sc = new Scanner(System.in);
        int a=sc.nextInt() ;
```

```
        int b=sc.nextInt();
        c=(a>b)?b:a;
        for(int i=c;i>1;i--){
            if(a%i==0 && b%i==0){
                System.out.println(i);
                break;
            }
        }
    }
}
```

48. Write a Java program to display the middle value of the string. If the string is odd then display the single element in the middle, if even then display two elements from the middle.

Input Format

Input to get a string.

Output Format

Display the middle element as shown in the sample output.

Sample Input 1:

carry

Sample Output 1:

r

Sample Input 2:

lemons

Sample Output 2:

mo

```
import java.util.*;
public class qn48 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        String st=in.nextLine();
        int mid=st.length()/2;
        if(st.length()%2==0){
            System.out.println(st.charAt(mid-1)+""+st.charAt(mid));
        }
        else{
            System.out.println(st.charAt(mid));
        }
    }
}
```

49. Given an arrangement of strings, the task is to find out the second most frequent string in the arrangement.

Note: No two strings are the second most repeated, there will be always a single string.

Input Format

The first line of input contains an integer X denoting number of strings in a sequence and the second line contains X space separated strings.

Output Format

Display the second most repeated string.

Constraints

3<=X<=103

1<=|String length|<=100

Sample Input

6

aaa bbb ccc bbb aaa aaa

Sample Output

bbb

Sample Input

6

geeks for geeks for geeks aaa

Sample Output

for

```java
import java.util.*;

public class qn49 {
    public static int find_len(String s,ArrayList<String> al){
        int count=0;
        for(String str : al){
            if(str.equals(s)){
                count++;
            }
        }
        return count;
    }
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int n=in.nextInt();
        in.nextLine();
        String s="";
        ArrayList<String> list = new ArrayList<>();
        for(int i=0;i<n;i++){
            s=in.next();
            list.add(s);
        }
        Set<String> set = new HashSet<>(list);

        for(int i=0 ;i<2;i++){
            int max=0;
            for(String st:set){
                if(qn49.find_len(st, list)>max){
                    max=qn49.find_len(st, list);
                    s=st;
                }
```

```
            }
            if(i==0){
                list.remove(s);
                set.remove(s);
            }
        }
        System.out.println(s);
    }
}
```

50. Banu, an overall coordinator of Independence Day celebration wants to arrange the freedom fighters photo in specific pattern.

If number of Freedom fighters photo is 5, she needs to arrange as shown in below sample.

Note : Freedom fighters photo are represented in alphabets in the sample output.

--------e--------

------e-d-e------

----e-d-c-d-e----

--e-d-c-b-c-d-e--

e-d-c-b-a-b-c-d-e

--e-d-c-b-c-d-e--

----e-d-c-d-e----

------e-d-e------

--------e--------

Input Format

Number of freedom fighters photo as integer in first line

Output Format

Output should display the suitable pattern

Sample Input

5

Sample Output

--------e--------

------e-d-e------

----e-d-c-d-e----

--e-d-c-b-c-d-e--

e-d-c-b-a-b-c-d-e

--e-d-c-b-c-d-e--

----e-d-c-d-e----

------e-d-e------

--------e--------

```
import java.util.*;
public class qn50 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int num=in.nextInt();
```

```
        ArrayList<String>list=new
ArrayList<>(Arrays.asList("abcdefghijklmnopqrstuvwxyz".split("")));
        ArrayList<Integer>al1=new ArrayList<>();
        ArrayList<Integer>al2=new ArrayList<>();

        for(int i=num-1;i>=0;i--){
            al1.add(i);
            if(i<num-1){
                al2.add(num-i-1);
            }
            int n1=1;
            for(int j=0;j<num;j++){
                if(al1.contains(j)){
                    System.out.print(list.get(num-n1));
                    if(j<num-1){
                        System.out.print("-");

                    }
                    n1++;
                }
                else{
                    System.out.print("--");
                }
            }
            int n2=al2.size();
            for(int j=1;j<num;j++){
                if(al2.contains(j)){
                    System.out.print("-"+list.get(num-n2));
                    n2--;
                }
                else{
                    System.out.print("--");
                }
            }
            System.out.println();
        }
        al1.remove((Integer)0);
        for(int i=1;i<=num-1;i++){
            al2.remove((Integer)i);
            int n1=1;
            for(int j=0;j<num;j++){
                if(al1.contains(j)){
                    System.out.print(list.get(num-n1));
                    if(j<num-1){
                        System.out.print("-");

                    }
                    n1++;
                }
                else{
```

```
                    System.out.print("--");
                }
            }
            int n2=al2.size();
            for(int j=0;j<=num-2;j++){
                if(al2.contains(num-j-1)){
                    System.out.print("-"+list.get(num-n2));
                    n2--;
                }
                else{
                    System.out.print("--");
                }
            }
            System.out.println();
            al1.remove((Integer)i);



        }
    }
}
```

51. Akash given you an integer **N**. You should convert the integer to binary format (32 bit) and flip all the bits(1-> 0 and 0-> 1) and print the result as an integer.

for example, **N** = 9, then

$00000000000000000000000000001001_2 = 9_{10}$

$11111111111111111111111111110110_2 = 4294967286_{10}$

You should print 4294967286. as a answer.

Input Format

First line of input contains an integer **N.**

Output Format

Output the required integer.

Constraints

$0 <= N <= 2^{32}$

Sample Input

Sample Output

1

4294967294

```
import java.util.*;

public class qn51 {
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int num = in.nextInt();
        String bit_32="00000000000000000000000000000000";
```

```
        String bin = "";
        while(num!=0){
            if(num%2==1){
                bin+=Integer.toString(1);
            }
            else{
                bin+=Integer.toString(0);
            }
            num/=2;
        }
        StringBuilder stb = new StringBuilder();
        stb.append(bin);
        bin =stb.reverse().toString();
        bit_32=bit_32.substring(0,bit_32.length()-bin.length());
        bit_32+=bin;
        bit_32=bit_32.replace("0", "9");
        bit_32=bit_32.replace("1", "0");
        bit_32=bit_32.replace("9", "1");
        stb.delete(0, stb.length());
        stb.append(bit_32);
        bin =stb.reverse().toString();
        long sum=0L;
        for(int i=0;i<bin.length();i++){
            String e=Character.toString(bin.charAt(i));
            if(e.equals("1")){
                sum+=Math.pow(2,i);
            }
        }
        System.out.println(sum);

    }
}
```

52. Smith is very much interested in sorting the arrays. One day, he asked his friend David to sort the arrays based on the decreasing frequency. David is well known about sorting, but he doesn't know about the decreasing frequency. so, help him to solve the problem.

Input Format

First line of input contains an integer **N** representing the size of the array

Next line consists of **N** space separated integers **A[i]**

Output Format

Print the sorted array in decreasing frequency

Constraints

$2 < N < 50$

$1 < A[i] < 100$

Sample Input

11

2 3 2 4 5 12 2 3 3 3 12

Sample Output

3 3 3 3 2 2 2 12 12 4 5

```java
import java.util.*;
public class qn52 {
    public static int find_len(String s,ArrayList<String> al){
        int count=0;
        for(String str : al){
            if(str.equals(s)){
                count++;
            }
        }
        return count;
    }
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int n=in.nextInt();
        in.nextLine();
        String s="";
        String result="";
        ArrayList<String> list = new ArrayList<>();
        for(int i=0;i<n;i++){
            s=in.next();
            list.add(s);
        }
        Set<String> set = new HashSet<>(list);
        int size=set.size();
        for(int i=0 ;i<size;i++){
            int max=0;
            for(String st:set){
                if(qn49.find_len(st, list)>max){
                    max=qn49.find_len(st, list);
                    s=st;
                }
            }
            for(int j=0 ; j<max;j++){
                result+=s+" ";
            }

            list.remove(s);
            set.remove(s);
            max=0;

        }
        System.out.println("\n"+result);

    }
}
```

53. Raju and Somu are playing a game in which there are pots of silver arranged in a line, each containing some silver coins. They get alternating turns in which the player can pick a pot from one of the ends of the line. The winner is the player who has a higher number of coins at the end. The objective is to maximize the number of coins collected by Raju, assuming Somu also plays

optimally. You are required to return an integer denoting the max coins Raju could get while playing the game. You may assume that Raju starts the game.

Input Format

First Line of input contains an integer denotng the number of silver pots.

Next line contains **N** space separated values denoting the values (arr[]) of silver coins in each pot.

Output Format

Print the maximum amount of money **X** can collect.

Constraints

1<=**N**<=100

1<=**arr[]**<=1000

Sample Input

4

8 15 3 7

Sample Output

22

```
import java.util.*;

public class qn53{
    public static String check(int i){
        if(i%2==0){
            return "even";
        }
        else{
            return "odd";
        }
    }
    public static void main(String arg[]){
        Scanner in = new Scanner(System.in);
        int num = in.nextInt();
        in.nextLine();
        String str = in.nextLine();
        ArrayList<String> list = new ArrayList<>(Arrays.asList(str.split(" ")));
        int max=0 , index=0,e;
        int sum=0,count=0;
        int size=list.size();
        size=(size%2==0)?size/2:(size/2)+1;
        String set="";
        for(int j=0 ;j<list.size();j++){
            if(count<size){
                for(int i=0 ;i<list.size();i++){
                    e=Integer.parseInt(list.get(i));
                    if(e>max){
                        max=e;
                        index=i;
                    }
                }
                if(j==0){
                    set=qn53.check(index);
                    sum+=max;
```

```
                    count++;


                }
                else if(j>0){
                    if(set.equals(qn53.check(index))){
                        sum+=max;
                        count++;

                    }
                }
                list.set(index,"0");
                max=0;
                index=0;
            }
            else{
                break;
            }

        }
        System.out.println(sum);
    }
}
```

54.. Ram gifted a new Vespa scooter to Jannu. But Jannu's father instructed Jannu to drive within speed limits. So every day Jannu's father asks the distance traveled(in km) and time taken(in hrs) to cover the distance. He calculates the speed using given values and check Jannu's speed.

Help Jannu's father to calculate the speed.

Note : Display the result in km/hr unit

Input Format

Distance traveled in first line

Time taken to travel in second line

Output Format

Speed with units as shown in sample output

Sample Input

200

10

Sample Output

20.00 km/hr

Sample Input

50.5

10

Sample Output

5.05 km/hr

Sample Input

90

4.5

Sample Output

20.00 km/hr

```java
import java.util.*;
class qn54
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        float a = sc.nextFloat();
        float b = sc.nextFloat();
        float c;
        c = a/b;
        System.out.println(c +" Km/hr");
    }
}
```

55.Design and write a class to represent a bank account that includes the following members:

- Data members
  - Owner name
  - Account number
  - Balance amount in the account
- Methods members
  - To assign initial values
  - To deposit an amount
  - To withdraw an amount after checking balance
  - To display the owner name and balance
- incorporate a constructor to provide initial values.
- Write a main method for the above class that reads in the initial values from the keyboard.

Sample Output:

Please Enter an Account Number: 1001

Please Enter the Account Holder Name: Dhaya

Please Enter the Balance: 120000

Account Holder Name: Dhaya

Account Balance: 120000.0

Account Holder Name: Dhaya

Account Balance: 240000.0

Error: Insufficient fund or Invalid amount!

Account Holder Name: Dhaya

Account Balance: 240000.0

Account Holder Name: Dhaya

Account Balance: 120000.0

```java
import java.io.*;

import java.util.*;
class BankAccount{
    private String name;
    private int num;
    private double blns;
    BankAccount(String s , int n , double b){
        name=s ;
        num= n;
        blns = b;
    }
    public void display(){
        System.out.println("Account Holder Name: "+name);
        System.out.println("Account Balance: "+blns);
```

```
    }
    public void withdraw(double w){
        if(w>=blns){
            System.out.println("Error: Insufficient fund or Invalid amount!");
        }
        else{
            blns=blns-w;
        }
    }
    public void deposit(double d){
        blns = blns+d;
    }
}
class qn55{
    public static void main(String a[]){
        Scanner obj = new Scanner(System.in);
        System.out.print("Please Enter an Account Number: ");
        int num = Integer.parseInt(obj.nextLine());
        System.out.print("please Enter the Account Holder Name: ");
        String name = obj.nextLine();
        System.out.print("Please Enter the Balance: ");
        double blns = Double.parseDouble(obj.nextLine());
        BankAccount acc = new BankAccount(name , num , blns);
        acc.display();
        acc.deposit(blns);
        acc.display();
        acc.withdraw(2*blns);
        acc.display();
        acc.withdraw(blns);
        acc.display();

    }
}
```

56. Create a new class called "City" that can be used to keep track of location information for a given city. Your class should include the following

1. String name
2. double lon (for longitude)
3. double lat (for latitude)
4. A constructor that accepts a name, lon and lat value and stores them in the instance variables for the object
5. A method that reports the current position of a city. Here's a method header to get you started:

   public void report()

6. A method that computes the distance from the lon and lat of one city to the lon and lat of another city. Use the standard distance formula to compute this value. Here's a method header to get you started:

public double distanceFrom(lon1, lat1, lon2, lat2)

Create a new class called "Ex1 2". Do the following in this class:

1. Include the "main" function
2. Create two objects for the two cities with name, longitude and latitude values
3. Calculate the distance between these two cities

Sample Output:

City #1
Name: NYC
Longitude: 50
Latitude: 75

City #2
Name: Chicago
Longitude: 25
Latitude: 10

City: NYC is at: 50.0, 75.0

City: Chicago is at: 25.0, 10.0

NYC is 7394 kms away from Chicago

```
class City{
    String name;
    double lon;
    double lat;
    City(String n,double lo , double la){
        name=n;
        lon = lo;
        lat = la;
    }
    public void report(){
        System.out.println("City: "+name+" is at: "+lon+", "+lat);
    }
    public double distanceFrom(double lon1 , double lat1 , double lon2 , double lat2){
        long R=6371L;
        double r1= Math.toRadians(lat1);
        double r2= Math.toRadians(lat2);
        double dla = Math.toRadians(lat2-lat1);
        double dlo = Math.toRadians(lon2-lon1);
        double a =
        Math.sin(dla/2)*Math.sin(dla/2)+Math.sin(dlo/2)*Math.sin(dlo/2)*Math.cos(r1)*Math.cos
(r2);
        double c = 2*Math.atan2(Math.sqrt(a),Math.sqrt(1-a));
        double d = R*c;
        return d;
    }
}
class qn56{
    public static void main(String args[]){
        City ob1= new City("NYC",50.0,75.0);
        City ob2 = new City("Chicago",25.0,10.0);
        System.out.println("City#1");
        System.out.println("Name: "+ob1.name);
        System.out.println("Longitude: "+(int)ob1.lon);
```

```
        System.out.println("Latitude: "+(int)ob1.lat);
        System.out.println();
        System.out.println("City#2");
        System.out.println("Name: "+ob2.name);
        System.out.println("Longitude: "+(int)ob2.lon);
        System.out.println("Latitude: "+(int)ob2.lat);
        System.out.println();
        ob1.report();
        System.out.println();
        ob2.report();
        System.out.println();
        int distance=(int)ob2.distanceFrom(ob1.lon,ob1.lat,ob2.lon,ob2.lat);
        System.out.println(ob1.name+" is "+distance+" kms away from "+ob2.name);

    }
}
```

57. Write a Java program to display a series of student ID numbers and ask the user to enter a test letter grade for the student. Create an Exception class named GradeException that contains a static public array of valid letter grades('A', 'B', 'C', 'D','E', 'F' and 'I'), which you can use to determine whether a grade entered from the application is valid. Create a method named validGrade() which throws a GradeException if the user does not enter a valid letter grade. Catch the GradeException, and display the appropriate message. If the user enters a valid grade then the corresponding student id and grade are updated in the Hashtable. At the end of the application, display all the student IDs and grades which are available in the Hashtable.

Input =

A
B
R
I
G

Output =

The Student ID is :101
Enter the grade: A
The Student ID is :102
Enter the grade: B
The Student ID is :103
Enter the grade: R
java.lang.Exception: Grade Exception
The Student ID is :104
Enter the grade: I
The Student ID is :105
Enter the grade: G
java.lang.Exception: Grade Exception
Key/Values in Hashtable are:
{104=I, 102=B, 101=A}

```
import java.io.*;
import java.util.*;
class GradeException{
    Hashtable<Integer , String> ht=new Hashtable<>();
    public static String grade[]=new String[7];
```

```
    static{
        grade[0]="A";
        grade[1]="B";
        grade[2]="C";
        grade[3]="D";
        grade[4]="E";
        grade[5]="F";
        grade[6]="I";
    }
    void validGrade(int id , String c)throws Exception{
        List <String> GradeList = new ArrayList<>(Arrays.asList(grade));
        if(GradeList.contains(c)){
            ht.put(id,c);
        }
        else throw new Exception("Grade Exception");

    }
    void display(){
        System.out.println("Key/Values in HasHtable are:\n"+ht);
    }
}
public class qn57{
    public static void main (String arg[])throws IOException{
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
        GradeException g = new GradeException();
        int ID[] = new int[5];
        String grd;
        for(int i=0;i<5;i++){
            ID[i]=i+101;
            System.out.print("The Student ID is :"+ID[i]+"\nEnter the grade: ");
            grd=in.readLine();
            try{
                g.validGrade(ID[i],grd);
            }
            catch(Exception e){
                System.out.println(e);
            }
        }
        g.display();
    }
}
```

58.

```
import java.util.Scanner;

public class Exception1 {

  public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);

        int num = 0;

        do {
              System.out.println("Enter a number between 1 and 10");
              num = scan.nextInt();
              if (num < 1 || num > 10)
              System.out.println("\nIllegal value, " + num + " entered.   Please try
again.");
           }while (num < 1 || num > 10);

              System.out.println("\nValue correctly entered! Thank you.");

     }
 }
```

To Do:

1. Type the program above and compile. Run and enter an integer between 1 and 10.
2. The program is requesting a number between 1 and 10. Run the program again and enter 5.5. Although this number is between 1 and 10, the program will abort. Examine the error message. You should see the word Exception, the method where the exception occurred (main), the class name of the exception (InputMismatchException), as well as the call stack listing the method calls.
3. Add a try/catch block to catch and handle the InputMismatchException exception. Identify the statements that cause the error as well as the portions of the program that depend upon these statements. Enclose these statements within the try block. Follow the try block with the catch block given below. Note, the InputMismatchException class is defined in java.util and must be imported. Also, when the Scanner throws an InputMismatchException, the input token will remain in the buffer so that it can be examined by the program. In our case, we will not be examining the token, but will simply clear out of the buffer to start over.

```
catch (InputMismatchException ime) {

  System.out.println("Enter whole numbers only, with no spaces or other characters");
  scan.next();
 }
```

4. Compile and run the program again, testing with a variety of input (integers, floats, characters) The program should not abort when floats or character data is given.

```
import java.util.Scanner;

public class qn58{
    public static void main(String arg[]){
       Scanner scan = new Scanner(System.in);
        int num = 0;
        do{
            System.out.println("Enter a number between 1 and 10");
            try{
                num= scan.nextInt();
                if (num < 1 || num>10){
                    System.out.println("\nIllegal value, "+ num +" entered. Please try
again.");
                }
            }
```

```
            catch (Exception ime){
                System.out.println("Enter whole numbers only, with no spaces or other
characters");
                scan.next();
            }
        }while(num<1 || num>10);

        System.out.println("\nValue correctly entered! Thank you.");
    }
}
```

59. Complete the below java program by incorporating appropriate method overloading.

```
public class Add{
//overloading methods
public static void main( String args[]){
Add ob = new Add();
//Read e1, e2 and e3 as Integer Input
//Read e4 and e5 as Double Input
ob.add(e1,e2);
ob.add(e1,e2,e3);
ob.add(e4 , e5);
}
}
```

```
class sample{
    public int add(Integer... i){
        int sum=0;
        for(int n:i){
            sum=sum+n;
        }
        return sum;
    }
    public double add(Double... i){
        double sum=0;
        for(double n:i){
            sum=sum+n;
        }
        return sum;
    }
}
public class qn59{
    public static void main (String arg[]){
        sample obj = new sample();
        int e1=1 , e2=2 , e3=3;
        double e4=9.3 , e5=6.1;
        System.out.println("Sum of Three integers: "+obj.add(e1,e2,e3));
        System.out.println("Sum of Two integers: "+obj.add(e3,e1));
```

```
        System.out.println("Sum of Two doubles: "+obj.add(e4,e5));
    }
}
```

60. write a `main` method in a different class to briefly experiment with some instances of the `Account` class.

- Using the `Account` class as a base class, write two derived classes called `SavingsAccount` and `CurrentAccount`. A `SavingsAccount` object, in addition to the attributes of an `Account` object, should have an interest variable and a method which adds interest to the account.

  A `CurrentAccount` object, in addition to the attributes of an `Account` object, should have an overdraft limit variable. Ensure that you have overridden methods of the `Account` class as necessary in both derived classes.
- Now create a `Bank` class, an object of which contains an array of `Account` objects. Accounts in the array could be instances of the `Account` class, the `SavingsAccount` class, or the `CurrentAccount` class. Create some test accounts (some of each type).
- Write an update method in the bank class. It iterates through each account, updating it in the following ways: Savings accounts get interest added (via the method you already wrote); CurrentAccounts get a letter sent if they are in overdraft.
- The `Bank` class requires methods for opening and closing accounts, and for paying a dividend into each account.

Hints:

- Note that the balance of an account may only be modified through the `deposit(double)` and `withdraw(double)` methods.
- The `Account` class should not need to be modified at all.
- Be sure to test what you have done after each step.

Output:

```
Savings Account Balance = 10000.0 Interest : 0.25
Current Account Balance = 20000.0 Limit : 1000.0
After updating the interest rate
Savings Account Balance = 10125.0 Interest : 1.25
After updating the withdrawn limit
Current Account Balance = 20000.0 Limit : 2000.0
Withdraw Rs. 1000 from Current Account
Current Account Balance = 19000.0 Limit : 2000.0
Withdraw Rs. 1000 from Current Account
Current Account Balance = 18000.0 Limit : 2000.0
Withdraw Rs. 3000 from Current Account
Sorry, the limit is exceeded
Current Account Balance = 18000.0 Limit : 2000.0
```

```
class Account{
    private double bal;
    private int accnum;

    Account(){}
    public Account(int a){
        bal=0.0;
        accnum=a;
    }

    public void deposit(double sum){
        if (sum>0){
            bal+=sum;
        }
        else System.err.println("Account.deposit(...): "+"cannot deposit negative amount.");

    }
```

```
    public void withdraw(double sum){
        if (sum>0) bal-=sum;
        else System.err.println("Account withdraw(...): "+"cannot withdraw negative
amount.");

    }
    public double getAccountNumber()
    {
        return accnum;

    }
    public double getBalance(){
        return bal;
    }
    public String toString(){
        return "Acc "+accnum+": " + "balance ="+bal;
    }
    public final void print()
    {
        System.out.println(toString());
    }
}
class SavingsAccount extends Account{
    double interest=0.0;
    public SavingsAccount(double b,double i){
        super.deposit(b);
        interest=i;
    }
    public void setInterest(double n){
        interest=n;
        super.deposit(interest*100);
        System.out.println("After updating the interest rate");
        print();
    }
    public String toString(){
        return "Savings Account Balance = "+getBalance()+" Interest : "+interest;
    }
}
class CurrentAccount extends Account{
    double limit=0.0;
    public CurrentAccount(double b , double l){
        super.deposit(b);
        limit=l;
    }
    public void setLimit(double l){
        limit = l;
        System.out.println("After updating the withdrawn limit");
        print();
    }
    public String toString(){
```

```
        return "Current Account Balance = "+getBalance()+" Limit : "+limit;
    }
    public void withdraw(double num){
        System.out.println("Withdraw Rs. "+(int)num+" from Current Account");
        if(num<=limit){
            super.withdraw(num);

            print();
        }
        else System.out.println("Sorry, the limit is exceeded");
    }
}
class qn60{
    public static void main(String arg[]){
        Account a;
        a = new Account(1920102080);
        SavingsAccount b= new SavingsAccount(10000.0,0.25);
        CurrentAccount c = new CurrentAccount(20000.0,1000.0);
        b.print();
        c.print();
        b.setInterest(1.25);
        c.setLimit(2000.0);
        c.withdraw(1000);
        c.withdraw(1000);
        c.withdraw(3000);
        c.print();
    }
}
```

61. Define an interface "IntOperations" with methods to check whether an integer is positive, negative, even, odd, prime and operations like factorial and sum of digits. Define a class MyNumber having one private int data member and implement the interface. Write a default constructor to initialize it to 0 and another constructor to initialize it to a value (Use this).

Hint:

```
public static void main(String a[])
   {
      MyNumber m=new MyNumber(11);
      m.integer();
      m.evenodd();
      m.prime();
      m.factorial();

       m.sumofdigit();
    }
```

Output:

```
11 is a Positive Number
11 is a Odd Number
11 is a Prime Number
The factorial of 11 is 39916800
Sum of it's digits is 2

14 is a Positive Number
14 is a Even Number
14 is not a Prime Number
The factorial of 14 is 1278945280
Sum of it's digits is 5
```

```java
interface IntOperations {
    void integer();
    void evenodd();
    void prime();
    void factorial();
    void sumofdigit();
}
class MyNumber implements IntOperations{
    public int n;
    MyNumber(){n=0;}
    MyNumber(int i){n=i;}
    public void integer(){
        if(n>0){
            System.out.println(n+" is a Positive Number");
        }
        else System.out.println(n+" is a Negative Number");
    }
    public void evenodd(){
        if(n%2==0){
            System.out.println(n+" is a Even Number");
        }
        else System.out.println(n+" is a Odd Number");
    }
    public void prime(){
        int h=n/2;
        boolean fg=false;
        for(int i=2 ; i<=h ; i++){
            if(n%i==0) fg=true;
        }
        if(fg) System.out.println(n+" is not a Prime Number");
        else System.out.println(n+" is a Prime Number");
    }
    public void factorial(){
        int f=1;
        for(int i=1;i<=n;i++){
            f*=i;
        }
        System.out.println("The factorial of "+n+" is "+f);
    }
    public void sumofdigit(){
        int temp=n,sum=0;
        while(temp !=0){
            sum+=temp%10;
            temp/=10;
        }
        System.out.println("Sum of it's digits is "+sum);
    }
}
public class qn61{
    public static void main(String arg[]){
```

```
        MyNumber m=new MyNumber(11);
        MyNumber m2=new MyNumber(14);
        m.integer();
        m.evenodd();
        m.prime();
        m.factorial();
        m.sumofdigit();
        System.out.println();
        m2.integer();
        m2.evenodd();
        m2.prime();
        m2.factorial();
        m2.sumofdigit();


    }
}
```

62. Define an interface "StackOperations" which declares methods for a static stack.

```
interface StackOperations{
        int max=5;
        void push(int data);
        void pop();
        int isempty();
        int isfull();
}
```

Define a class "MyStack" which contains an array and top as data members andimplements the above interface. Initialize the stack using a constructor. Write a menudriven program to perform operations on a stack object.
Hint: Menudriven

```
import java.io.*;
interface StackOperations{
    int max=5;
    void push(int data);
    void pop();
    int isempty();
    int isfull();
}
class MyStack implements StackOperations{
    public int arr[]=new int[max];
    public int pos=max;
    public int isempty(){
        if(pos==max){
            return 1;
        }
        else return 0;
    }
    public int isfull(){
        if(pos==0){
            return 1;
        }
```

```
        else return 0;
    }
    public void push(int data){

            pos--;
            arr[pos]=data;

        }

    public void pop(){

            arr[pos]=0;
            pos++;
        }

    public void display(){
        for(int i=pos;i<max;i++){
            System.out.println(arr[i]);
        }
    }
}
public class qn62{
    public static void main(String arg[])throws Exception{
        int ch,data;
        String c;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        MyStack s = new MyStack();
        do{
            System.out.println("\n1:Push");
            System.out.println("\n2:Pop");
            System.out.println("\n3:Display");
            System.out.println("\n4:Exit");
            System.out.println("\nEnter your choice:");
            ch=Integer.parseInt(br.readLine());
            switch(ch){
                case 1:
                    if(s.isfull()==1){
                        System.out.println("Stack is full");

                    }
                    else{
                        System.out.println("Enter the data:");
                        data = Integer.parseInt(br.readLine());
                        s.push(data);
                    }break;
                case 2:
                    if(s.isempty()==1){
                        System.out.println("Stack is empty");

                    }
```

```
                    else{

                        s.pop();
                    }break;
                case 3:
                    if(s.isempty()==1){
                        System.out.println("Stack is empty");
                    }
                    else{
                        System.out.println("The Elements in the Stack are:");
                        s.display();
                    }break;
                case 4:
                    System.exit(0);
                    break;
                default:
                    System.out.println("\nInvalid choice");
                }
            }while(ch!=4);

        }
    }
```

63. Write a JAVA program that creates threads by extending Thread class . Firstthread display "Good Morning " every 1 sec, the second thread displays "Hello "every 2 seconds and the third display "Welcome" every 3 seconds.

```
class GoodMorning extends Thread {
    public void run() {
        try {
            int i=0;
            while (i<5) {
                sleep(1000);
                System.out.println("Good morning ");
                i++;
            }
        } catch (Exception e) {
        }
    }
   }
class Hello extends Thread {
    public void run() {
        try {
            int i=0;
            while (i<5) {
                sleep(2000);
                System.out.println("hello");
                i++;
            }
```

```
        } catch (Exception e) {
        }
    }
}
class Welcome extends Thread {
    public void run() {
        try {
            int i=0;
            while (i<5) {
                sleep(3000);
                System.out.println("welcome");
                i++;
            }
        } catch (Exception e) {
            }
    }
}
class qn63{
    public static void main(String args[]) {
        GoodMorning t1 = new GoodMorning();
        Hello t2 = new Hello();
        Welcome t3 = new Welcome();
        t1.start();
        t2.start();
        t3.start();

    }
}
```

64. Write a JAVA program that creates threads by implementing Runnable . Firstthread display "Good Morning " every 1 sec, the second thread displays "Hello "every 2 seconds and the third display "Welcome" every 3 seconds.

```
class Frst implements Runnable {
    Thread t;
    Frst() {
        t = new Thread(this);
        System.out.println("Good Morning");
        t.start();
    }
    public void run() {
        for (int i = 0; i < 10; i++) {
            System.out.println("Good Morning");
            try {
                t.sleep(1000);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
```

```
}
class sec implements Runnable {
    Thread t;
    sec() {
        t = new Thread(this);
        System.out.println("hello");
        t.start();
    }
    public void run() {
        for (int i = 0; i < 10; i++) {
            System.out.println("hello");
            try {
                t.sleep(2000);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}
class third implements Runnable {
    Thread t;
    third() {
        t = new Thread(this);
        System.out.println("welcome");
        t.start();
    }
    public void run() {
        for (int i = 0; i < 10; i++) {
            System.out.println("welcome");
            try {
                t.sleep(3000);
            } catch (Exception e) {
                System.out.println(e);
            }
        }
    }
}
public class qn64{
    public static void main(String arg[]) {
        new Frst();
        new sec();
        new third();
    }
}
```

65. Create three Classes: Storage, Counter and Printer. The Storage class should store an integer. The Counter class should create a thread that starts from counting from 0(0, 1, 2, 3....) and stores each value in the Storage class. The Printer class should create a thread that keeps reading the value in the Storage class and printing it. Create a Java program that create an instance of the

Storage class, and sets up a Counter and a Printer object to operate on it. Modify the program to ensure that each number is printed exactly once, by adding suitable synchronization

```
class Storage{
    int i=0;
    boolean value=false;
    public synchronized void add(int i) throws InterruptedException {
        this.i=i;
        this.value=true;
    }
    public synchronized int display() throws InterruptedException {
        this.value=false;
        return i;
    }
}
class Counter extends Thread{
    Storage s;
    Counter(){}
    Counter(Storage s){this.s=s;}
    public void run() {
        for(int i=0 ;i<10;i++) {
            try {
                synchronized(s) {
                    while(s.value==true) {
                        s.wait();
                    }
                    s.add(i);
                    System.out.println("Added: "+i);
                    s.notifyAll();
                }
            } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            }
        }
    }
}
class Printer extends Thread{
    Storage s;
    Printer(){}
    Printer(Storage s){this.s=s;}
    public void run() {
        for(int i=0 ;i<10;i++) {
        try {
            synchronized(s) {
                while(s.value==false) {
                    s.wait();
                }
                System.out.println("Print: "+s.display());
```

```
                s.notifyAll();
            }
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
            }
        }
    }
}
public class qn65{
    public static void main(String arg[]) {
        Storage s1= new Storage();
        Counter c= new Counter(s1);
        Printer p = new Printer(s1);
        try {
            c.start();
            p.start();
        }
        catch(Exception E) {
            System.out.println(E);
        }
    }
}
```

66. A string is said to be complete if it contains all the characters from **a** to **z**. Given a string, check if it complete or not.

Input
First line of the input contains the number of strings **N**. It is followed by **N** lines each contains a single string.

Output
For each test case print "YES" if the string is complete, else print "NO"

Constraints
1 <= N <= 10
The length of the string is at max **100** and the string contains only the characters **a** to **z**

SAMPLE INPUT

```
3
wyyga
qwertyuioplkjhgfdsazxcvbnm
ejuxggfsts
```

SAMPLE OUTPUT

```
NO
YES
NO
```

```
import java.util.*;
```

```java
public class qn66{
    public static void main(String arg[]){
        System.out.println("number of terms :");
        int n;
        Scanner in = new Scanner(System.in);
        n=in.nextInt();
        System.out.println("\n");
        System.out.println("imput string is :");
        String str;
        in.nextLine();
        ArrayList<String> al = new ArrayList<String>();
        for (int u =0 ; u<n ; u++ ){
            str=in.nextLine();
            al.add(str);
        }
        System.out.println("\n");
        System.out.println("program output:");
        String arr[]
={"a","b","c","d","e","f","g","h","i","j","k","l","m","n","o","p","q","r","s","t","u","v","w"
,"x","y","z"};
        boolean flag=false;
        for (String g: al){
            for (String k: arr){
                if(g.contains(k)){
                    flag=true;
                }
                else{
                    flag=false;
                    break;
                }
            }

            if(flag){
                System.out.println("YES");
            }
            else{
                System.out.println("NO");
            }
        }
    }
}
```

67. Sometimes it's better to use dynamic size arrays. Java's Arraylist can provide you this feature. Try to solve this problem using Arraylist.

You are given lines. In each line there are zero or more integers. You need to answer a few queries where you need to tell the number located in x^{th} position y^{th} of line.

Take your input from System.in.

Input Format

The first line has an integer . In each of the next lines there will be an integer denoting number of integers on that line and then there will be space-separated integers. In the next line there will be an integer denoting number of queries. Each query will consist of two integers and.

Constraints

- 1<=n<=20000
- 0<=d<=50000
- 1<=q<=1000
- 1<=x<=n

Each number will fit in signed integer.
Total number of integers in lines will not cross .

Output Format

In each line, output the number located in x^{th} position of y^{th} line. If there is no such position, just print "ERROR!"

Sample Input

```
5
5 41 77 74 22 44
1 12
4 37 34 36 52
0
3 20 22 33
5
1 3
3 4
3 1
4 3
5 5
```

Sample Output

```
77
36
4
ERROR!
ERROR!
```

```java
import java.util.*;
public class qn67 {
    public static void main(String[] args) {
        ArrayList<ArrayList> numList = new ArrayList<ArrayList>();
        int n=0;
        Scanner in = new Scanner(System.in);
        n=in.nextInt();

        for(int k =0 ; k<n ; k++){
            ArrayList<Integer> na = new ArrayList<Integer>();
            int nr = in.nextInt();
            na.add(nr);
            for(int j =0 ; j<nr ; j++){
                int elm = in.nextInt();
                na.add(elm);
            }
            numList.add(na);
        }
        n=in.nextInt();

        for(int k =0 ; k<n ; k++){
```

```
            try {
                int x=in.nextInt();
                int y=in.nextInt();
                System.out.println(numList.get(x-1).get(y-1));
            } catch (Exception e) {
            //TODO: handle exception
                System.out.println("ERROR!");
            }
        }
    }
}
```

68. Create an interface called "employee". The employee interface contains the method called "getInput()" to get the name, employee number, employee designation and monthly salary. Create another interface called compensation. This new interface should contain a method which returns a value of type double data item called incentive, by taking two inputs of float type called the period and compensation to calculate the incentive for the overtime duty he/she has seen. Create new classes called manager, scientist, and laborer, and use interface methods for getting the inputs from these three classes and calculate the incentives for each type of employee. Get all the details of each type of employee and print the details along with the incentive amount which is calculated from the members of the two interfaces.

```
import java.util.*;
interface employee{
    public void getInput();
}

interface compensation{
    public double getIncentive( float period,float compensation);
}
class Manager implements employee ,compensation {

    String name;
    String number;
    String Designation;
    double Monthly_salary;
    float period,compensation;
    Scanner scan = new Scanner(System.in);
    @Override
    public double getIncentive(float p,float c) {
        double d=p*c;
        return d;
    }

    @Override
    public void getInput() {
        // TODO Auto-generated method stub
        System.out.println("\nEnter Manager Details");
        System.out.println("Enter Manager Name: ");
        this.name=scan.nextLine();
        System.out.println("Enter Manager Number: ");
        this.number=scan.nextLine();
```

```
        System.out.println("Enter Manager Designation: ");
        this.Designation=scan.nextLine();
        System.out.println("Enter Manager Monthly Salary: ");
        this.Monthly_salary=scan.nextDouble();
        System.out.println("\nEnter over time details:");
        System.out.println("Enter over time period:");
        this.period=scan.nextFloat();
        System.out.println("Enter over time compensation:");
        this.compensation=scan.nextFloat();
        System.out.println("Incentive Amount:"+getIncentive(this.period, this.compensation));
    }

    public void print(){
        System.out.println("\tManager Details");
        System.out.println("Manager Name: "+this.name);
        System.out.println("Manager Number: "+this.number);
        System.out.println("Manager Designation: "+this.Designation);
        System.out.println("Manager Monthly Salary: "+this.Monthly_salary);
        System.out.println("Incentive Amount:"+getIncentive(this.period, this.compensation));

    }

}
class Scientist implements employee ,compensation{

    String name;
    String number;
    String Designation;
    double Monthly_salary;
    float period,compensation;
    Scanner scan = new Scanner(System.in);
    @Override
    public double getIncentive(float p,float c) {
        // TODO Auto-generated method stub
        double d=p*c;
        return d;
    }

    @Override
    public void getInput() {
        // TODO Auto-generated method stub
        System.out.println("\nEnter Scientist Details");
        System.out.println("Enter Scientist Name: ");
        this.name=scan.nextLine();
        System.out.println("Enter Scientist Number: ");
        this.number=scan.nextLine();
        System.out.println("Enter Scientist Designation: ");
        this.Designation=scan.nextLine();
        System.out.println("Enter Scientist Monthly Salary: ");
        this.Monthly_salary=scan.nextDouble();
```

```
        System.out.println("\nEnter over time details:");
        System.out.println("Enter over time period:");
        this.period=scan.nextFloat();
        System.out.println("Enter over time compensation:");
        this.compensation=scan.nextFloat();
        System.out.println("Incentive Amount:"+getIncentive(this.period, this.compensation));

    }

    public void print(){
        System.out.println("\tScintist Details");
        System.out.println("Scientist Name: "+this.name);
        System.out.println("Scientist Number: "+this.number);
        System.out.println("Scientist Designation: "+this.Designation);
        System.out.println("Scientist Monthly Salary: "+this.Monthly_salary);
        System.out.println("Incentive Amount:"+getIncentive(this.period, this.compensation));

    }

}
class Laborer implements employee ,compensation{

    String name;
    String number;
    String Designation;
    double Monthly_salary;
    float period,compensation;
    Scanner scan = new Scanner(System.in);
    @Override
    public double getIncentive(float p,float c) {
        double d=p*c;
        return d;
    }

    @Override
    public void getInput() {
        // TODO Auto-generated method stub
        System.out.println("\nEnter Laborer Details");
        System.out.println("Enter Laborer Name: ");
        this.name=scan.nextLine();
        System.out.println("Enter Laborer Number: ");
        this.number=scan.nextLine();
        System.out.println("Enter Laborer Designation: ");
        this.Designation=scan.nextLine();
        System.out.println("Enter Laborer Monthly Salary: ");
        this.Monthly_salary=scan.nextDouble();
        System.out.println("\nEnter over time details:");
        System.out.println("Enter over time period:");
        this.period=scan.nextFloat();
        System.out.println("Enter over time compensation:");
```

```
        this.compensation=scan.nextFloat();
        System.out.println("Incentive Amount:"+getIncentive(this.period, this.compensation));

    }

    public void print(){
        System.out.println("\tLaborer Details");
        System.out.println("Laborer Name: "+this.name);
        System.out.println("Laborer Number: "+this.number);
        System.out.println("Laborer Designation: "+this.Designation);
        System.out.println("Laborer Monthly Salary: "+this.Monthly_salary);
        System.out.println("Incentive Amount:"+getIncentive(this.period, this.compensation));

    }
}


public class qn68 {

    public static void main(String arg[]){
        Manager manager = new Manager();
        Scientist scientist=new Scientist();
        Laborer laborer = new Laborer();

        manager.getInput();
        scientist.getInput();
        laborer.getInput();

        manager.print();
        scientist.print();
        laborer.print();

    }

}
```

69. Write a Java Program to read a file which contains numbers and find the minimum and maximum.

Sample Input:
data1.txt
Sample Output:
min = 1
max = 1
Sample Input: (If it is empty file)
data2.txt
Sample Output:
Empty array
min undefined
max undefined

```
import java.io.*;
import java.util.*;
class SortedList{
    ArrayList <Integer> array = null;
    SortedList(){
        array = new ArrayList<Integer>();
    }
    public void add(int u ){
        array.add(u);
        Collections.sort(array);
    }
    public boolean isEmpty(){
        return array.isEmpty();
    }
    public int getFirst(){
        return array.get(0);
    }
    public int getLast(){
        return array.get(array.size()-1);
    }
}
public class qn69{
    public static void main(String arg[]){
        String data = null ;
        Scanner sc = new Scanner (System.in);
      data=sc.nextLine();
        try{
            File fi = new File(data.trim());
            FileReader fr = new FileReader(fi);
            BufferedReader dip = new BufferedReader(fr);
            String i;
            SortedList sl = new SortedList();
            while((i=dip.readLine())!=null){
                sl.add(Integer.parseInt(i));
            }
            if(sl.isEmpty()){
```

```
                System.out.println(" Empty array");
                System.out.println("min undefined");
                System.out.println("max undefined");
            }
            else{
                System.out.println(" min = "+ sl.getFirst());
                System.out.println("max = "+ sl.getLast());
            }
            fr.close();
        }
        catch(Exception e){
            System.out.println(e);
        }
    }
}
```