FS PROGRAMS

**29/03/2022**

/\*Write a Java Program to Find the Non-Repeated Elements in an Array

Example:

input=

8

12 25 8 12 43 5 8 43

output=

25 5

\*/

import java.util.\*;

class Test

{

public static void main (String args[])

{

Scanner sc = new Scanner (System.in);

int n = sc.nextInt ();

int a[] = new int[n];

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

{

a[i] = sc.nextInt ();

}

int count=0;

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

count=0;

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

if(a[i]==a[j]){

count++;

}

}

if(count==1){

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

}

}

}}

/\*Write a Java program

Create an Employee class to read employee details like Empno, name and

basic salary then calculate gross salary of an employee with respect to

specified conditions mentioned below and then print basic details along with gross.

If Basic salary<=10k then HRA=20%;DA=80%

If Basic salary<=20k then HRA=25%;DA=90%

If Basic salary>20k then HRA=30%;DA=95%

Gross salary= Basic salary + HRA+ DA

Sample:

Input=

Enter the employee id:101

Enter the employee name:Varun

Enter the basic salary of an employee:25000

Output=

Id: 101

Name: Varun

Basic salary: 25000

Gross salary: 56250.0

\*/

import java.util.\*;

class GrossSalary{

public static void main(String[]args)

{

Scanner sc=new Scanner(System.in);

System.out.print("Enter the employee id:");

int empid = sc.nextInt();

System.out.print("Enter the employee name:");

String name = sc.next();

System.out.print("Enter the basic salary of an employee:");

int sal = sc.nextInt();

float hra, da, gsal;

if(sal<=10000)

{

hra=sal\*20/100;

da=sal\*80/100;

gsal=sal+hra+da;

System.out.println("Id: "+empid);

System.out.println("Name: "+name);

System.out.println("Basic salary: "+sal);

System.out.println("Gross salary: "+gsal);

}

else if(sal<=20000)

{

hra=sal\*25/100;

da=sal\*90/100;

gsal=sal+hra+da;

System.out.println("Id: "+empid);

System.out.println("Name: "+name);

System.out.println("Basic salary: "+sal);

System.out.println("Gross salary: "+gsal);

}

else if(sal>20000)

{

hra=sal\*30/100;

da=sal\*95/100;

gsal=sal+hra+da;

System.out.println("Id: "+empid);

System.out.println("Name: "+name);

System.out.println("Basic salary: "+sal);

System.out.println("Gross salary: "+gsal);

}

}

}

/\*Write a Java program to check whether the given non-negative number is SPYNUMBER or not.

A SPYNUMBER is a number where the sum of its digits equals the product of its digits.

Ex: 1124 is SPYNUMBER, because the sum of its digits(1+1+2+4) = 8 and the product of its digits (1\*1\*2\*4) = 8 is equal. If the number is SPYNUMBER print true else print false.

Sample :

input=1124

output=true

input=1113

output=false

input=-5

output=-1\*/

import java.util.\*;

class SpyNumber{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int sum=0,mul=1;

if(n<0){

System.out.println("-1");

}

else{

while(n>0){

int digit=n%10;

sum=sum+digit;

mul=mul\*digit;

n=n/10;

}

if(sum==mul){

System.out.println("true");

}

else{

System.out.println("false");

}}

}

}

**30/03/2022**

/\*Write a Java program to accept a number from user and count zero, odd and even digits

of the entered number.

Sample:

Input=

Enter an integer

12045067

Output=

Zeros:2

Evennumbers:3

Oddnumbers:3 \*/

import java.util.\*;

class oddEvenZero{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("Enter an integer");

String str=sc.nextLine();

String b[]=str.split("");

int a[]=new int[b.length];

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

a[i]=Integer.parseInt(b[i]);

}

int zero=0,even=0,odd=0;

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

if(a[i]==0){

zero++;

}

else if(a[i]%2==0){

even++;

}

else if(a[i]%2!=0) {

odd++;

}

}

System.out.println("Zeros:"+zero);

System.out.println("Evennumbers:"+even);

System.out.println("Oddnumber:"+odd);

}

}

/\* Write a Java Program to check whether the entered number is Automorphic Number or not.

Description- A number is called Automorphic if and only if its square ends

in the same digits as the number itself.

Sample 1:

input=5

output=Yes

Sample 2:

input=14

output=No \*/

import java.util.\*;

class Automorphic{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("Enter a number");

int n=sc.nextInt();

int square=n\*n;

if(square%10==n%10){

System.out.println("Yes");

}

else{

System.out.println("No");

}

}

}

/\*Anik, a student of mathematics is doing a research for his PhD degree.

He thinks if he can complete his work, it will help mankind a lot.

But the problem is as part of his research he needs to check sum of two numbers in a special way.

The numbers themselves will be big enough plus the sum needs to be done in a reversed way.

For example, if two numbers were 24 and 13, we have to first reverse these numbers to make them 42 and 31 and then we have to do the addition so the sum will be 42 + 31 = 73.

Afterwards he needs to output the result as reversed again, so the output will be 37 actually.

One advantage he has is that the input numbers will never have leading zero.

But the resulted sum if has any leading zeroes need to be removed.

For example,

if we have two numbers such as 911 and 199, then the reversed numbers will be 119 and 991 and the sum will be 119 + 991 = 1110.

So, if we reverse, the output should be 0111, but actually we have to consider 111 as output because we should remove all leading zeroes in final output.

As it is hard for Anik, so he wants to hire a freelancer who can write a small program to help him.

Sample:

input=

199

911

output=

111

input=19

22

output=311

\*/

import java.util.\*;

class SumReverse{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int number1=sc.nextInt();

int number2=sc.nextInt();

int num1=reverse(number1);

int num2=reverse(number2);

int sum=reverse(num1+num2);

System.out.println(sum);

}

public static int reverse(int number){

int reverse = 0;

while(number != 0)

{

int remainder = number % 10;

reverse = reverse \* 10 + remainder;

number = number/10;

}

return reverse;

}

}

**31/01/2022**

/\*Diamond pattern\*/

import java.util.\*;

class Diamond

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

System.out.println("How Many Rows You Want In Your Diamond");

int noOfRows = sc.nextInt();

int midRow = noOfRows/2;

int row = 1;

for (int i = midRow; i > 0; i--)

{

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

{

System.out.print(" ");

}

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

{

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

}

System.out.println();

row++;

}

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

{

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

{System.out.print(" "); }

for (int j = row; j > 0; j--)

{

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

}

System.out.println();

row--;

}

}}

/\* perfect squares with divisors\*/

import java.util.\*;

class PerfectSquares{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int count=0;

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

if(n%i==0){

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

if(sqrt\*sqrt==i){

count++;

}

}

}

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

}

}

/\* ugly numbers\*/

import java.util.\*;

class UglyNumber{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int count=0;

while(n!=1){

if(n%5==0){

n=n/5;

}

else if(n%2==0){

n=n/2;

}

else if(n%3==0){

n=n/3;

}

else{

System.out.println("false");

count=1;

break;

}

}

if(count==0){

System.out.println("true");

}

}

}

/\* string balanced parenthis \*/

import java.util.\*;

public class Test {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

while(sc.hasNext()) {

String input = sc.next();

System.out.println(balanaced(input));

}

}

public static boolean balanaced(String s) {

Stack<Character> stack = new Stack<Character>();

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

char c = s.charAt(i);

if(c =='[' || c == '(' || c == '{') {

stack.push(c);

}else if(c == ']') {

if(stack.isEmpty() || stack.pop() != '[') {

return false;

}

}else if(c == ')') {

if(stack.isEmpty() || stack.pop() != '(') {

return false;

}

}else if(c == '}') {

if(stack.isEmpty() || stack.pop() != '{') {

return false;

}}}

return stack.isEmpty();

}}

**04/04/2022**

/\* In a tag-based language like XML or HTML, contents are enclosed between a start tag

and an end tag like <tag>contents</tag>. Note that the corresponding end tag starts

with a /.

Given a string of text in a tag-based language, parse this text and retrieve the

contents enclosed within sequences of well-organized tags meeting the following

criterion:

The name of the start and end tags must be same. The HTML code <h1>Hello World</h2>

is not valid, because the text starts with an h1 tag and ends with a non-matching h2

tag. Tags can be nested, but content between nested tags is considered not valid.

For example, in <h1><a>contents</a>invalid</h1>, contents is valid but invalid is

not valid. Tags can consist of any printable characters.

Input Format

The first line of input contains a single integer, N (the number of lines).

The N subsequent lines each contain a line of text.

Output Format

For each line, print the content enclosed within valid tags.

If a line contains multiple instances of valid content, print out each

instance of valid content on a new line; if no valid content is found, print None.

sample case

input

4

<h1>Nayeem loves counseling</h1>

<h1><h1>Sanjay has no watch</h1></h1><par>So wait for a while</par>

<Amee>safat codes like a ninja</amee>

<SA premium>Imtiaz has a secret crush</SA premium>

output

Nayeem loves counseling

Sanjay has no watch

So wait for a while

None

Imtiaz has a secret crush \*/

import java.util.Scanner;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Test{

public static void main(String[] args){

Scanner scan = new Scanner(System.in);

int testCases = Integer.parseInt(scan.nextLine());

while (testCases-- > 0) {

String line = scan.nextLine();

boolean matchFound = false;

Pattern r = Pattern.compile("<(.+)>([^<]+)</\\1>");

Matcher m = r.matcher(line);

while (m.find()) {

System.out.println(m.group(2));

matchFound = true;

}

if ( ! matchFound) {

System.out.println("None");

}

}

}

}

/\* Write a java program .Given a number n, write an efficient function to print all unique prime factors of n.

For example, if the input number is 12, then output should be "2 3".

And if the input number is 315, then output should be "3 5 7".

Sample Input:

315

Sample Output:

3 5 7 \*/

import java.util.\*;

class UniquePrimeFactors

{

public static void main(String args[])

{

Scanner g=new Scanner(System.in);

int n=g.nextInt();

for(int i=2;i<=n;i++)

{

if(n%i==0)

{

int count=0;

for(int j=2;j<=i-1;j++)

{

if(i%j==0)

{

count++;

break;

}

}

if(count==0)

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

}

}

}

}

/\*Write a Java program to print the Largest digit of a given positive number.

Sample 1:

input=

12534

output=

5

Sample 2:

input=

100

output=

1 \*/

import java.util.\*;

class LargestDigit{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int count=0;

while(n>0){

int digit=n%10;

n=n/10;

if(count<digit){

count=digit;

}

}

System.out.println(count);

}

}

/\*

Write a java program that takes two arguments: the original price and the discount percentage

as integers and returns the final price after the discount (Integer).

Alternative Text

Examples sample

input output

1500, 50---> 750

89, 20---> 71

100, 75---> 25

Sample:

input=150

50

output=75 \*/

import java.util.\*;

class Price{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int original=sc.nextInt();

int discount=sc.nextInt();

int s=100-discount;

int finalprice=(original\*s)/100;

System.out.println(finalprice);

}

}

/\* create a class with common attributes sno,sname and a constructor to initialize them, display method to display their values. create a sub class mcastudent with attributes ds,co,java and a constructor to initialize them, display method to display their values. create a sub class mbastudent with attributes hr,finance,marketing and a constructor to initialize them, display method to display their values.

Output

1

shiva

77

99

88

2

kiran

88

66

99 \*/

import java.util.\*;

class student

{

int sno;

String sname;

student(int sno,String sname)

{

this.sno=sno;

this.sname=sname;

}

void display()

{

System.out.println(sno);

System.out.println(sname);

}

}

class mcastudent extends student

{

int da,co,java;

mcastudent(int sno,String sname,int da,int co,int java)

{

super(sno,sname);

this.da=da;

this.co=co;

this.java=java;

}

void display()

{

super.display();

System.out.println(da);

System.out.println(co);

System.out.println(java);

}

}

class mbastudent extends student

{

int hr,finance,marketing;

mbastudent(int sno,String sname,int hr,int finance,int marketing)

{

super(sno,sname);

this.hr=hr;

this.finance=finance;

this.marketing=marketing;

}

void display()

{

super.display();

System.out.println(hr);

System.out.println(finance);

System.out.println(marketing);

}

}

class Constructor1

{

public static void main(String args[])

{

mcastudent m=new mcastudent(1,"shiva",77,99,88);

m.display();

mbastudent n=new mbastudent(2,"kiran",88,66,99);

n.display();

}

}

**06/04/22**

/\*input:

How Many Rows You Want In Your Pyramid

9

output:

9

8 9 8

7 8 9 8 7

6 7 8 9 8 7 6

5 6 7 8 9 8 7 6 5

4 5 6 7 8 9 8 7 6 5 4

3 4 5 6 7 8 9 8 7 6 5 4 3

2 3 4 5 6 7 8 9 8 7 6 5 4 3 2

1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1 \*/

import java.util.\*;

class Pattern2

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("How Many Rows You Want In Your Pyramid");

int n=sc.nextInt();

int count=1;

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

{

for(int j=1;j<i\*2;j++)

{

System.out.print(" ");

}

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

{

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

}

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

{

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

}

System.out.println();

count++;

}

}

}

/\*Write a program that prints the numbers from 1 to n and for multiples of '3' print "Fizz"

instead of the number, for the multiples of '5' print "Buzz" and

for multiples of '15' print FizzBuzz.

Sample:

input:

maximum range of input

20

output=

1

2

Fizz

4

Buzz

Fizz

7

8

Fizz

Buzz

11

Fizz

13

14

FizzBuzz

16

17

Fizz

19

Buzz

\*/

import java.util.\*;

class FizzBuzz{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("maximum range of input");

int n=sc.nextInt();

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

if(i%3==0 && i%5!=0){

System.out.println("Fizz");

}

else if(i%5==0 && i%3!=0){

System.out.println("Buzz");

}

else if(i%3==0 && i%5==0){

System.out.println("FizzBuzz");

}

else{

System.out.println(i);

}

}

}

}

/\*Write a java program to check whether the entered number is binary number or not

Sample 1:

input=111100

output=true

Sample 2 :

input=1120

output=false \*/

import java.util.\*;

class BinaryChecking{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

String str=sc.next();

int count=0;

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

if(str.charAt(i)=='1' || str.charAt(i)=='0'){

count++;

}

}

if(count==str.length()){

System.out.println("true");

}

else{

System.out.println("false");

}

}

}

/\*Write a Java program to check whether a given positive number is a Harshad Number or not.

Description: A harshad number in a given number base, is an integer that is divisible by the sum of its digits when written in that base.

Example: Number 200 is a Harshad Number because the sum of digits 2 and 0 and 0 is 2(2+0+0) and 200 is divisible by 2.

Number 171 is a Harshad Number because the sum of digits 1 and 7 and 1 is 9(1+7+1) and 171 is divisible by 9.

Harshad Number is also called as Niven Number.

Sample 1:

input=200

output=Yes

Sample 2:

input=101

output=No

Sample 3:

input=-12

output=-1

Note : if a given number is negative print -1. \*/

import java.util.\*;

class HarshadNumber{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int sum=0;

int t=n;

if(n<0){

System.out.println("-1");

}

else{

while(n>0){

int digit=n%10;

sum=sum+digit;

n=n/10;

}

if(t%sum==0){

System.out.println("Yes");

}

else{

System.out.println("No");

}}

}

}

**07/04/2022**

/\*Write a java program to find if a given positive number is a Camel Case Number or Not

A number is said to be Camel Case Number if the First digit of the number is always greater than

or equal to the digits which are right side to it and if a number contains a single digit it is

always assumed it as a Camel Case. If the given Number is Camel Case Print "Camel Case" else print "Not Camel Case" If any negative number is given print "Not Camel Case"

sample 1:

input=

543

output=

Camel Case

sample 2:

input=

582

output=

Not Camel Case \*/

import java.util.\*;

class CamelCaseNumber

{

public static void main(String args[])

{

Scanner cc=new Scanner(System.in);

int Num=cc.nextInt();

int Reminder=0,Maxinum=0;

if(Num<0)

{

System.out.println("Not Camel Case");

}

else

{

while(Num>0)

{

Reminder=Num%10;

if(Maxinum< Reminder)

{

Maxinum=Reminder;

}

Num=Num/10;

if(Reminder==Maxinum)

{

System.out.println("Camel Case");

}

else

{

System.out.println("Not Camel Case");

}

}

}

}

}

Other method:

import java.util.\*;

public class CamelCaseNumber{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

String s=sc.next();

String[] arr=s.split("");

boolean result=true;

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

{

for(int j=i+1;j<arr.length;j++)

{

if(Integer.valueOf(arr[i])<Integer.valueOf(arr[j]))

{

result=false;

break;

}

}

}

if(result)

System.out.print("Camel Case");

else

System.out.print("Not Camel Case");

}

}

/\* Write a java program to check if a given positive number is a strong number.

A strong number is a number whose sum of factorial of each digit is the number itself.

Print 1 if it is strong number, else 0.

example: 145

1! + 4! + 5! = 1+24+120 = 145

Sample:

Input=145

output=1

Input=128

output=0 \*/

import java.util.\*;

class StrongNumber

{

public static void main(String args[])

{

Scanner sn=new Scanner(System.in);

int n=sn.nextInt();

int t=n,factSum=0;

while(t>0)

{

int fact=1;

int digit=t%10;

t=t/10;

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

{

fact=fact\*i;

}

factSum=factSum+fact;

}

if(n==factSum)

{

System.out.println("1");

}

else

{

System.out.println("0");

}

}

}

/\* Write a java program to find given number is perfect power or not

Given a positive integer which fits in a 32 bit signed integer,

find if it can be expressed as A^P where 10 > P > 1 and A > 0.

A and P both should be integers.

Sample input/output

input=125

output=True

explanation

as 5^3 = 125.

input=5

output=False \*/

import java.util.\*;

class PerfectPower{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int flag=0;

for(int p=2;p<10;p++){

for(int a=2;a<p;a++){

if(Math.pow(p,a)==n){

flag=1;

}

}

}

if(flag==1){

System.out.println("True");

}

else{

System.out.println("False");

}

}

}

/\* input:

How Many Rows You Want In Your Pyramid

9

output:

1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 8 7 6 5 4 3 2 1

1 2 3 4 5 6 7 6 5 4 3 2 1

1 2 3 4 5 6 5 4 3 2 1

1 2 3 4 5 4 3 2 1

1 2 3 4 3 2 1

1 2 3 2 1

1 2 1

1 \*/

import java.util.\*;

class Pattern3

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("How Many Rows You Want In Your Pyramid");

int n=sc.nextInt();

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

{

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

{

System.out.print(" ");

}

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

{

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

}

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

{

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

}

System.out.println();

}

}

}

**11/04/2022**

/\*Write a Java program to perform Decimal to any base Conversion.

Read two integers, First integer indicates decimal value and

second integer indicates base. After conversion print the output.

Note: If base<=0 or base>=9 print -1.

Sample 1:

input=10

2

output=1010

Sample 2:

input=100

5

output=400 \*/

import java.util.\*;

class Decimal\_AnyBase{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int decimal=sc.nextInt();

int base=sc.nextInt();

int index=0;

int a[]=new int[40];

int temp=decimal;

while(temp>0){

a[index]=temp%base;

temp=temp/base;

index++;

}

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

System.out.print(a[i]);

}

}

}

/\*Write a Java program to print following pattern

input=5

output=

55555

54444

54333

54322

54321 \*/

import java.util.\*;

class Pattern4

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

for(int a=n;a>=1;a--)

{

for(int b=n;b>=1;b--)

{

if(a>=b)

{

System.out.print(a);

}

else

{

System.out.print(b);

}

}

System.out.println();

}

}

}

/\*Write a program to check if a given number is COMPLETE NUMBER or NOT

A Number is said to be COMPLETE NUMBER if the individual digits of the number are all even

if the individual digits of the number are not even then print NOT COMPLETE NUMBER

Assume 0 as a Even Number

input = 486

output = COMPLETE NUMBER

input = 135

output = NOT COMPLETE NUMBER

input = 100

output = NOT COMPLETE NUMBER \*/

import java.util.\*;

class CompleteNumber{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int a=0;

int count=0;

while(n>0){

a++;

int remainder=n%10;

n=n/10;

if(remainder%2==0){

count++;

}

}

if(count==a){

System.out.println("COMPLETE NUMBER");

}

else{

System.out.println("NOT COMPLETE NUMBER");

}

}

}

/\*This Triangular Number Sequence is generated from a pattern of dots that form a triangle.

The first 5 numbers of the sequence, or dots, are:

1, 3, 6, 10, 15

This means that the first triangle has just one dot, the second one has three dots,

the third one has 6 dots and so on.

Write a function that gives the number of dots for its corresponding

triangle number of the sequence.

Examples:

input=1

output=1

input=6

output=21

input=215

output=2322 \*/

import java.util.\*;

class Dots{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int sum=0;

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

sum=sum+i;

}

System.out.println(sum);

}

}

/\*Write a Program in Java to check whether given number is a Disarium Number or not.

Disarium Number: A number is called Disarium if sum of its digits powered with their

respective positions is equal to the number itself.

For example 135 is a DISARIUM

(Workings 1^1+3^2+5^3 = 135)

Sample :

Input= 135

Output= Yes

Input=80

Output=No \*/

import java.util.\*;

class Disarium{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

String str=Integer.toString(n);

int len=str.length();

int t=n,sum=0;

while(t>0){

int digit=t%10;

t=t/10;

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

len--;

}

if(sum==n){

System.out.println("Yes");

}

else{

System.out.println("No");

}

}

}

**12/04/2022**

/\*Write a Java program to print following pattern

input=5

output=

10101

01010

10101

01010

10101 \*/

import java.util.\*;

class Pattern5{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

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

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

if((i+j)%2==0){

System.out.print("1");

}

else{

System.out.print("0");

}

}

System.out.println();

}

}

}

/\*Write a Java program to print following pattern

input=4

output=

1

23

456

78910 \*/

import java.util.\*;

class Pattern6{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int number=1;

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

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

System.out.print(number);

number++;

}

System.out.println();

}

}

}

/\*You hired three programmers and you (hopefully) pay them.

Write a java programe that takes three numbers (the hourly wages of each programmer) and

returns the difference between the highest-paid programmer and the lowest-paid.

Examples

input output

147, 33, 526 ------------> 493

33, 72, 74 ------------> 41

1, 5, 9 ------------> 8

Notes

Don't forget to print the result.

sample:

input=72

75

33

output=42 \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int a[]=new int[3];

int max=a[0],min=a[0];

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

a[i]=sc.nextInt();

}

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

if(a[i]>max){

max=a[i];

}

else {

min=a[i];

}

}

int diff=max-min;

System.out.println(diff);

}

}

/\* Write a Java program with the following Scenario

given some integers calcuate the average and check whether the average is whole

number or not

if it is whole number print --> true

else print --> false

Examples

2 //number of integers to be passed

1, 3-------->true

explanation: avg of 1,3 is 4/2 --> 2 so true

4

1, 2, 3, 4-------->false

explanation: avg of 1,2,3,4 is 10/4 --> 2.5 so false

Sample 1

input=2

1

3

output=true

Sample 2

input=4

1

2

3

4

output=false \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int sum=0;

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

int k=sc.nextInt();

sum=sum+k;

}

float avg=(float)sum/n;

int s=(int)avg;

if(avg==s){

System.out.println("true");

}

else{

System.out.println("false");

}

}

}

/\* Write a java program that takes three integers a, b, c and returns

the number of integers which are of equal value. (wrong solution)

Examples

3, 4, 3 -----> 2

1, 1, 1 -----> 3

3, 4, 1 -----> 0

Notes

Your function must return 0, 2 or 3.

Sample:

case=1

input=3

5

3

output=2 \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int a[]=new int[3];

int i=0,j=0;

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

a[i]=sc.nextInt();

}

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

int count=1;

for(j=i+1;j<a.length;j++){

if(a[i]==a[j]){

count++;

}

}

if(count>1)

System.out.println(count);

else if(count<1){

System.out.println("0");

}

}

}

}

**13/04/2022**

/\*Write a Java program to print following pattern

input=5

output=

ABCDEDCBA

ABCD DCBA

ABC CBA

AB BA

A A \*/

import java.util.\*;

class Pattern9{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

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

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

System.out.print((char)(j+64));

}

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

System.out.print(" ");

}

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

if(j!=n){

System.out.print((char)(j+64));

}

}

System.out.println();

}

}

}

/\*Write a Java program to print following pattern

input=5

output=

1

3 1

5 3 1

7 5 3 1

9 7 5 3 1 \*/

import java.util.\*;

class Pattern7{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

for(int i=1;i<n\*2;i=i+2){

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

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

}

System.out.println();

}

}

}

-----------------------------------------------------------------------------------------------------------------------------------

/\* Write a java program for the following scenario

Count the number of ones in the binary representation of an integer.

For example, since 12 is 1100 in binary,

the return value should be 2 (count of 1's in binary number.

Sample input /output

0 ---> 0

100 ---> 3

999 ---> 8

Sample:

input=196

output=3 \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int decimal=sc.nextInt();

int index=0;

int a[]=new int[40];

int temp=decimal;

while(temp>0){

a[index]=temp%2;

temp=temp/2;

index++;

}

int count=0;

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

if(a[i]==1){

count++;

}}

System.out.println(count);

}

}

/\*Write a Java program to print following pattern

input=5 output=

\*\*\*\*\*

\* \*

\* \*

\* \*

\*\*\*\*\* \*/

import java.util.\*;

class Pattern8{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

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

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

if(i==1 || i==n || j==1 || j==n){

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

}

else{

System.out.print(" ");

}

}

System.out.println();

}

}

}

/\*Write a Java program to print following pattern

input=5

output=

1

123

12345

1234567

123456789 \*/

import java.util.\*;

class Pattern10{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

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

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

System.out.print(" ");

}

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

System.out.print(j);

}

System.out.println();

}

}

}

**18/04/2022**

/\*Write a java program to print following pattern. Read the 10 random integers from user, store them in array and print them in right angled triangle pattern.

Sample :

input=

enter array elements

2 5 3 8 9 4 7 25 33 41

output=

2

5 3

8 9 4

7 25 33 41 \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("enter array elements");

int a[]=new int[10];

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

int n=sc.nextInt();

a[i]=n;

}

int c=0;

for(int i=1;i<=4;i++){

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

System.out.print(a[c]+" ");

c++;

}

System.out.println();

}

}

}

/\* Write a java program to print the following pattern for any given positive integer n

Sample:

input = 5

output =

0

505

45054

3450543

234505432

12345054321 \*/

import java.util.\*;

class Pattern11{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n = sc.nextInt();

System.out.println("0");

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

{

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

{

System.out.print(j);

}

System.out.print("0");

for (int k = n; k >= i; k--)

{

System.out.print(k);

}

System.out.println();

}

}

}

/\* The Employee class has two instance variables name,age. create a parameterized constructor to initialize them & display method to display the instance variables.

hint: create ArrayList to store objects of Employee class.

create and add First Employee object with values name-jack,age-35.

create and add Second Employee object with values name-steve,age-40.

Then display them by using the advanced for loop.

output:

jack

35

steve

40 \*/

import java.util.\*;

class Employee{

String name;

int age;

Employee(String name,int age){

this.name=name;

this.age=age;

}

void display(){

System.out.println(name);

System.out.println(age);

}

}

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

ArrayList<Employee> al=new ArrayList<>();

al.add(new Employee("jack",35));

al.add(new Employee("steve",40));

for(Employee e:al){

e.display();

}

}

}

/\*Write a Java program to find the duplicate values of an array of integer values.

Print all the duplicate values in the array if have else print -1.

test case=1

input=

enter size

6

enter elements

1

6

4

6

5

1

output=

1 6

test case=2

input=

enter size

5

enter elements

1

2

3

4

5

output=

output=

-1 \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("enter size");

int n=sc.nextInt();

System.out.println("enter elements");

int a[]=new int[n];

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

a[i]=sc.nextInt();

}

int count=0;

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

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

if(a[i]==a[j]){

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

count++;

}

}

}

if(count==0){

System.out.print("-1");

}

}

}

/\*You are given a list of N transfers (numbered from 0 to N.1) between two banks,

bank A and bank B. The K-th transfer is described by two values

- R[K] (either "A" or "B") representing the recipient

(the bank the transfer is sent to);

- V[K] denoting the value sent via the transfer.

All transfers are completed in the order they appear on the list. The banks

do not want to go into debt (i.e. their account balance may not drop below 0).

What minimum initial account balance in each bank is necessary in order to

complete the transfers?

Write a function:

-----------------

public int[] solution(String R, int[] V);

that, given a string R and an array of integers V, both of length N, returns

an array of two integers. The integers should represent the minimum

initial account balances for banks A and B in the following order:

[bank A, bank B].

Result array should be returned as an array of integers.

Input Format:

-------------

Line-1: A String R, consist of letters A nd B only of length N

Line-2: N space separated integers, array V[].

Output Format:

--------------

Print the initial bank balances of bank-A abd bank-B

Samlpe Input-1:

---------------

BAABA

2 4 1 1 2

Sample Output-1:

----------------

2 4

Explanation:

------------

Given R = "BAABA" and V = [2, 4, 1, 1, 2], the function should return[2, 4].

The bank accounts balances after each transfer are shown in the following table

|A B

============================

initial balance |2 4

transfer 2 from A to B |2 6

transfer 4 from B to A |4 2

transfer 1 from B to A |5 1

transfer 1 from A to B |4 2

transfer 2 from B to A |6 0

Samlpe Input-2:

---------------

ABAB

10 5 10 5

Sample Output-2:

----------------

0 15

Explanation:

------------

Given R = "ABAB" and V = [10, 5, 10, 15], the function should return [0, 15]

case=1

input=

BAABA

2 4 1 1 2

output=

2 4

case=2

input=

ABAB

10 5 10 5

output=

0 15

case=3

input=

B

100

output=

100 0

case=4

input=

ABBABBBAA

2 5 6 9 6 5 4 4 2

output=

15 2

case=5

input=

BAABBAAAABBBABBABAAA

34 21 36 10 28 12 20 26 19 36 9 4 34 38 33 24 26 7 7 7

output=

34 62

case=6

input=

BABABAABBBBBBBBAABBBAABAAAAABAAABAAAAABBBBAABBBBAB

35 5 1 16 13 12 3 12 4 7 26 37 30 9 32 10 14 7 3 21 19 37 36 27 17 29 4 34 30 12 21 3 13 25 20 40 34 24 39 12 9 21 21 2 9 8 39 15 20 11

output=

177 90 \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

String str=sc.nextLine();

int size=str.length();

int arr[]=new int[size];

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

arr[i]=sc.nextInt();

}

int minA=0,minB=0,a=0,b=0;

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

if(str.charAt(i)=='B'){

b+=arr[i];

a-=arr[i];

}

else{

b-=arr[i];

a+=arr[i];

}

minA=Math.min(minA,(a));

minB=Math.min(minB,(b));

}

System.out.println(-minA+" "+-minB);

}

}

**19/04/2022**

//Q2) Java Program to Find the Largest Number in an Array

/\* Sample Input & Output:

Enter number of elements in the array:

5

Enter elements of array:

4

2

3

6

1

Maximum value:6 \*/

import java.util.\*;

class Largest\_Number{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("Enter number of elements in the array:");

int n=sc.nextInt();

int a[]=new int[n];

System.out.println("Enter elements of array:");

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

a[i]=sc.nextInt();

}

int max=a[0];

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

if(a[i]>max){

max=a[i];

}

}

System.out.println("Maximum value:"+max);

}}

//Java Program to Insert an Element in a Specified Position in a Given Array

/\* Sample Input & Output:

Enter no. of elements you want in array:6

Enter all the elements:

2

4

6

8

9

11

Enter the position where you want to insert element:2

Enter the element you want to insert:55

After inserting:2,55,4,6,8,9,11 \*/

import java.util.\*;

class Insert\_Array{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.print("Enter no. of elements you want in array:");

int n=sc.nextInt();

int a[]=new int[n+1];

System.out.println("Enter all the elements:");

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

a[i]=sc.nextInt();

}

System.out.print("Enter the position where you want to insert element:");

int position=sc.nextInt();

System.out.print("Enter the element you want to insert:");

int element=sc.nextInt();

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

a[i+1]=a[i];

}

a[position-1]=element;

System.out.print("After inserting:");

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

System.out.print(a[i]+",");

}

}

}

/\*Q1) Java Program to Increment Every Element of the Array by One & Print Incremented Array\* /\*

Sample Input & Output:

output:

Enter no. of elements you want in array:

5

Enter all the elements:

2

5

8

6

9

Elements of array after increment by 1:

3

6

9

7

10

\*/

import java.util.\*;

class Increment\_Array{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("Enter no. of elements you want in array:");

int n=sc.nextInt();

int a[]=new int[n];

System.out.println("Enter all the elements:");

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

a[i]=sc.nextInt();

}

System.out.println("Elements of array after increment by 1:");

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

a[i]=a[i]+1;

System.out.println(a[i]);

}

}

}

---------------------------------------------------------------------------------------------------------------------------------

/\*Write a Java program to merge two arrays into the third array.

Sample:

enter size of 1st array

5

enter size of 2nd array

4

enter elements into 1st array

32

36

35

54

45

enter elements into 2nd array

69

65

78

98

after merging

32 36 35 54 45 69 65 78 98 \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("enter size of 1st array");

int n1=sc.nextInt();

System.out.println("enter size of 2nd array");

int n2=sc.nextInt();

int a[]=new int[n1];

int a1=a.length;

System.out.println("enter elements into 1st array");

for(int i=0;i<a1;i++){

a[i]=sc.nextInt();

}

int b[]=new int[n2];

int b1=b.length;

System.out.println("enter elements into 2nd array");

for(int i=0;i<b1;i++){

b[i]=sc.nextInt();

}

int c1=a1+b1;

int c[]=new int[c1];

for(int i=0;i<a1;i++){

c[i]=a[i];

}

for(int i=0;i<b1;i++){

c[a1+i]=b[i];

}

System.out.println("after merging");

for(int i=0;i<c1;i++){

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

}

}

}

/\*Java Program to Delete the Specified Integer from an Array

sample input and output

Enter no. of elements you want in array:

6

Enter all the elements:

6

3

8

9

1010

2

Enter the element you want to delete:

1010

After Deleting:

6

3

8

9 \*/

import java.util.\*;

class Delete{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("Enter no. of elements you want in array:");

int n=sc.nextInt();

int a[]=new int[n];

System.out.println("Enter all the elements:");

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

a[i]=sc.nextInt();

}

System.out.println("Enter the element you want to delete:");

int element=sc.nextInt();

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

if(a[i]==element){

for(int j=i;j<a.length-1;j++){

a[j]=a[j+1];

}

break;

}

}

System.out.println("After Deleting:");

for(int i=0;i<a.length-1;i++){

System.out.println(a[i]);

}

}

}

**21/04/2022**

/\*Write a Java program to check whether a number is a Duck Number or not.

A Duck number is a number which has zeroes present in it, but there should be no zero present in the beginning of the number.

Sample

input=3210

output=true

input=0234

output=false

input=1234

false \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int temp=n;

int count=0;

while(temp>0){

int digit=temp%10;

temp=temp/10;

if(digit==0){

count++;

}

}

if(count==1){

System.out.println("true");

}

else{

System.out.println("false");

}

}

}

/\*

1 2 3 4 5 6 7 8 9

2 3 4 5 6 7 8 9 1

3 4 5 6 7 8 9 1 2

4 5 6 7 8 9 1 2 3

5 6 7 8 9 1 2 3 4

6 7 8 9 1 2 3 4 5

7 8 9 1 2 3 4 5 6

8 9 1 2 3 4 5 6 7

9 1 2 3 4 5 6 7 8

\*/

import java.util.\*;

class Pattern20{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=9;

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

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

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

}

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

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

}

System.out.println();

}

}

}

/\*Write a Java program to check if two numbers are anagrams or not.

If Anagram Number print "Yes" else print "No".

Description: A number is said to be an anagram of some other number

if it can be made equal to the other number by just shuffling the digits in it.

Sample 1:

input=123

321

output=Yes

Sample 2:

input=121

113

output=No \*/

import java.util.\*;

class Anagram{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

String str1=sc.nextLine();

String str2=sc.nextLine();

char c1[]=str1.toCharArray();

char c2[]=str2.toCharArray();

Arrays.sort(c1);

Arrays.sort(c2);

if(Arrays.equals(c1,c2)){

System.out.println("Yes");

}

else{

System.out.println("No");

}

}

}

/\* Write a Java program to find the second most frequent character in a given string.

Sample:1

Input:

Enter the String:

success

Output:

The given string is: success

The second most frequent char in the string is: c

Sample:2

Input:

Enter the String:

worries

Output:

The given String is: worries

The second most frequent char in the string is: e \*/

import java.util.\*;

class Test

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter the String:");

String str=sc.nextLine();

System.out.println("The given string is: "+str);

int i,max=0,sec\_max=0;

int arr[]=new int[256];

for(i=0;i<str.length();i++)

{

if(str.charAt(i)!=' ')

arr[str.charAt(i)]++;

}

for (i = 0; i <256; i++) {

if (arr[i]>arr[max]) {

sec\_max=max;

max = i;

}

else if(arr[i]>arr[sec\_max] && arr[i]!=arr[max]){

sec\_max=i;

}

}

System.out.print("The second most frequent char in the string is: "+(char)(sec\_max));

}

}

/\*Java program to find the number of charters in the longest word in the sentence.

Sample I/O:

Enter any String:

I am an intern at MicrosoftSolutions

18

\*/

import java.util.\*;

class Test{

public static void main( String[] args )

{

Scanner sc=new Scanner(System.in);

System.out.println("Enter any String:");

String s = sc.nextLine();

String word[] = s.split( " " );

String rts = " ";

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

{

if ( word[i].length() > rts.length() )

rts = word[i];

}

System.out.println( rts.length() );

}}

/\*Write a java program to divide string into n equal parts

Sample Input n Output:

case=1

Enter any string:

good morning

three equal parts of given string are

good

mor

ning

case=2

Enter any string:

hello students

Sorry this string cannot be divided into Three equal parts.

case=3

input=Good Morning

three

output=

Enter any string:

three equal parts of given string are

Good

Mor

ning

\*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("Enter any string:");

String str=sc.nextLine();

int len=str.length();

int n=3;

int divide=len/3;

int temp=0;

String a[]=new String[n];

if(len%n!=0){

System.out.println("Sorry this string cannot be divided into Three equal parts.");

}

else{

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

String parts=str.substring(i,i+divide);

a[temp]=parts;

temp++;

}

System.out.println("three equal parts of given string are");

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

System.out.println(a[i]);

}}

}

}

/\*Aadhya wants to draw n lines, she puts in each line zero or more integers.

Aadhya wants you to answer few queries where you need to tell the number

located in yth position of xth line.

Take your input from System.in

Input Format

The first line has an integer "n". In each of the next "d" lines there will be an integer "d"

denoting number of integers on that line and then there will be space-separated integers.

In the next line there will be an integer "q" denoting number of queries.

Each query will consist of two integers "x" and "y" .

Output Format

In each line, output the number located in yth position of xth line.

If there is no such position, just print "ERROR!"

case=1

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

output=74

52

37

ERROR!

ERROR!

\*/

import java.util.\*;

public class Test {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

ArrayList arr[] = new ArrayList[n];

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

{

int size = sc.nextInt();

arr[i] = new ArrayList();

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

{

arr[i].add(sc.nextInt());

}

}

int q = sc.nextInt();

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

{

int x = sc.nextInt();

int y = sc.nextInt();

try

{

System.out.println(arr[x-1].get(y-1));

}

catch(Exception e)

{

System.out.println("ERROR!");

}

} }

}

/\* Write a java program for the following scenario

find the longest common prefix string amongst group of strings.

If there is no common prefix, return -1.

Example 1:

Input: 3 // no of strings

"flower","flow","flight" // strings do not use "" or , while reading

Output: "fl"

Example 2:

Input: 3 // number of strings

"dog","racecar","car" // strings do not use "" or , while reading

Output: -1

Explanation: There is no common prefix among the input strings.

Note;

strings consists of only lower-case English letters.

case=1

input=3

kmit

kmiengineering

kmcse

output=km

case=2

input=4

test

ting

treat

try

output=t

case=3

input=2

kmit

ngit

output=-1

\*/

(50 % passed)

import java.util.\*;

class Test {

public static void main(String args[]) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

String S[]=new String[n];

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

S[i]=sc.next();

}

String prefix = S[0];

for (int i = 1; i < S.length; i++)

while (S[i].indexOf(prefix) != 0) {

prefix = prefix.substring(0, prefix.length() - 1);

}

System.out.println(prefix);

}

}

**23/04/2022**

/\*Shyam is playing with group of characters, where the group of characters contain only A/B.

He has to return the count of non-empty substrings from this group which will have the same number of A's and B's and all the A's and all the B's in these substrings are grouped consecutively

Substrings that occur multiple times are counted the number of times they occur.

case 1:

Input = AABBAABB

Output = 6

Explanation:

There are 6 substrings that have equal number of consecutive B's and A's:

"AABB", "AB", "BBAA", "BA", "AABB", and "AB".

Notice that some of these substrings repeat and are counted the number of times they occur.

Also, "AABBAABB" is not a valid substring because all the A's (and B's) are not grouped together.

case 2:

Input = BABAB

Output = 4

Explanation:

There are 4 substrings: "BA", "AB", "BA", "AB" that have equal number of consecutive B's and A's.

test cases:

case=3

input=ABABABAB

output=7

case=4

input=AAAABBBB

output=4

case=5

input=ABA

output=2

case=6

input=BABABABAAABBB

output=10

case=7

input=AB

output=1 \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

String str=sc.nextLine();

int res=0,pre=0,cur=1;

for(int i=1;i<str.length();i++){

if(str.charAt(i)!=str.charAt(i-1)){

res+=Math.min(pre,cur);

pre=cur;

cur=1;

}

else{

cur++;

}

}

res+=Math.min(pre,cur);

System.out.println(res);

}

}

/\* sample input and output:

How many rows you want in this pattern

5

Here is your pattern

1

2 6

3 7 10

4 8 11 13

5 9 12 14 15 \*/

import java.util.\*;

class Pattern21{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("How many rows you want in this pattern");

int n=sc.nextInt();

System.out.println("Here is your pattern");

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

int number=i;

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

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

number=number+n-j;

}

System.out.println();

}

}

}

/\*Write a java program to reverse each word in a sentence using reverse method in StringBuilder class.

Sample I/O:

Enter any String:

Kindly concentrate on ur Studies

yldniK etartnecnoc no ru seidutS \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("Enter any String:");

String str=sc.nextLine();

String words[]=str.split("\\s");

String reverseWord="";

for(String w:words){

StringBuilder sb=new StringBuilder(w);

sb.reverse();

reverseWord+=sb.toString()+" ";

}

System.out.println( reverseWord.trim());

}

}

/\*Remove all digits & SP characters

2 Write a java program to remove all the digits and special characters from the

3 input string and display the output in the specified format.

4

5 Sample Output:"Pro!gra@mm&ing2Language"

6 String after removing special characters: ProgrammingLanguage

7 \*/

import java.util.\*;

class Test{

public static void main(String[]args)

{

String str="Pro!gra@mm&ing2Language";

String resultStr="";

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

{

if(str.charAt(i)>64 && str.charAt(i)<=122)

{

resultStr=resultStr+str.charAt(i);

}

}

System.out.println("String after removing special characters: "+resultStr);

}

}

**25/04/2022**

/\*Write a java program to find and print number of words in a line

and also print the words along with its length

Sample input:

Enter any String:

Welcome to kmit

3 words

\*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("Enter any String:");

String str=sc.nextLine();

int count=1;

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

if(str.charAt(i)==' '){

count++;

}

}

System.out.println(count+" words");

}

}

/\* Write a java program to print the missing element in array.

User gives the mean value of array elements then find the missing element and print it.

Note : always read size-1 elements.

Sample :

input=5 //size of array

10 20 30 50 //array elements

30 //mean value

output=

40 //missing element

\*/

import java.util.\*;

class Test

{

public static void main(String[]args)

{

Scanner s=new Scanner(System.in);

int n=s.nextInt();

int a[]=new int[n-1];

for(int i=0;i<n-1;i++)

{

a[i]=s.nextInt();

}

float mean=s.nextFloat();

int total=(n\*(n+1))/2;

int sum=0;

for(int i=0;i<n-1;i++)

{

sum=sum+a[i];

}

System.out.println((int)((mean\*n)-sum));

}

}

/\* Bob is working on Number Strings. He got an idea to find the smallest possible number by

deleting some digits from the number without changing the relative order of digits in it.

You will be given a integer String 'num', and an integer n.

Find the smallest number possible after deleting n digits from 'num'.

Note: If the number string 'num' turns to empty, print 0.

Input Format:

-------------

Line-1 : A string num, consist of digits only.

Line-2 : An integer n, number of digits to delete.

Output Format:

--------------

Print the smallest possible number.

Sample Input-1:

---------------

1432219

3

Sample Output-1:

----------------

1219

Explanation:

------------

Delete the three digits 4, 3, and 2 to form the smallest number 1219.

Sample Input-2:

---------------

10200

1

Sample Output-2:

----------------

200

Explanation:

------------

Delete the leading 1 and the smallest number is 200.

Note that the output must not contain leading zeroes. \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

String num=sc.nextLine();

int k=sc.nextInt();

System.out.println(Remove(num,k));

}

public static String Remove(String num,int k){

Stack<Character> st = new Stack<>();

int i=0;

for(i=0;i<num.length() && k>0;){

while(k>0 && !st.isEmpty() && num.charAt(i)< st.peek()){

st.pop();

k--;

}

st.push(num.charAt(i));

i++;

}

while(!st.isEmpty() && k>0){

st.pop();

k--;

}

String ans = "";

while(!st.isEmpty()){

ans = st.pop() + ans;

}

ans = ans + num.substring(i, num.length());

// 0002000

i = 0;

while(i<ans.length()){

if(ans.charAt(i) == '0'){

i++;

} else {

break;

}

}

// "000000"

return ans.substring(i).length() == 0 ? "0" : ans.substring(i);

}

}

Another solution

import java.util.\*;

public class Test

{

public static String removeKdigits\_Better(String num, int k) {

int len = num.length();

if(k==len)

return "0";

Stack<Character> stack = new Stack<>();

int i =0;

while(i<num.length()){

while(k>0 && !stack.isEmpty() && stack.peek()>num.charAt(i)){

stack.pop();

k--;

}

stack.push(num.charAt(i));

i++;

}

while(k>0){

stack.pop();

k--;

}

StringBuilder sb = new StringBuilder();

while(!stack.isEmpty())

sb.append(stack.pop());

sb.reverse();

while(sb.length()>1 && sb.charAt(0)=='0')

sb.deleteCharAt(0);

return sb.toString();

}

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

String input = sc.next();

int k = sc.nextInt();

System.out.println(removeKdigits\_Better(input, k));

}

}

**26/04/2022**

/\*There are N persons in the room, each person is paired with other person except one.

Every person is identified with an ID number, and every pair will have same ID number.

You will be given the ID's of N persons as a list[].

Your task is to find the person ID, who is not paired with anyone in the room.

Can you solve this problem in O(N) time complexity?

Input Format:

-------------

Line-1: An odd integer N, number of people.

Line-2: N space separated integers, the list[].

Output Format:

--------------

Print an integer result.

Sample Input-1:

---------------

5

5 3 4 5 3

Sample Output-1:

----------------

4

Sample Input-2:

---------------

7

8 2 6 8 3 2 6

Sample Output-2:

----------------

3 \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int a[]=new int[n];

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

a[i]=sc.nextInt();

}

int temp =0;

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

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

if(a[i]<a[j]){

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

} }

int count=0;

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

//int k;

for(int k =0; k<a.length;k++) {

if(a[j] == a[k]) {

count++;

}

}

if(count==1){

System.out.println(a[j]);

break;

}

count = 0;

}

}

}

/\*Neeraj wants to create a coding system which converts a word(all uppercase)

into a number as follows.

A means 1,

B means 2,

...

Z means 26,

AA means 27,

AB means 28,

...

AAA means 703

...

You are given a word W,

Your task is to help Neeraj to convert W into a number using

the given coding system.

Input Format:

-------------

A string W,a word.

Output Format:

--------------

Print an integer result.

Constraints:

---------

- 1 <= word.length <= 7

- W is in the range ["A", "FXSHRXW"].

Sample Input-1:

---------------

D

Sample Output-1:

----------------

4

Sample Input-2:

---------------

RRR

Sample Output-2:

----------------

12654 \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

String str=sc.next();

int result=0;

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

result = 26\*result + str.charAt(i) - 'A' + 1;

}

System.out.println(result);

}

}

/\*Write a java program to sort the strings in lexicographical order

Sample

input=

4

work hard dream big

output=

big dream hard work \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

String a[]=new String[n];

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

a[i]=sc.next();

}

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

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

if (a[i].compareTo(a[j]) > 0) {

String temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

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

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

}

}

}

/\*Write a Java program to return a new string where the last two characters

of a given string are swapped.

Sample:

input=string

output=strign

Note: If length of stringis less than to print the given string \*/

import java.util.\*;

class Test

{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

String str=sc.nextLine();

if(str.length()<2){

System.out.println(str);

}

else{

String a=str.substring(0, str.length()-2)+ str.charAt(str.length()-1) + str.charAt(str.length()-2);

System.out.println(a);

}

}

}

**27/04/2022**

/\*A number is called self-supportive if all the digits of the number are factors

of the number. For example, 48 is a self-supportive number because 48 % 4 == 0, and 48 % 8 == 0.

A number is not a self-supportive if it has any digit as zero.

Given two Positive numbers start and end, return a set of all the self-supportive

numbers in between start and end (both inclusive).

1<=start<=end<=10^4

Input Format:

-------------

Line: 2 space seperated integers start and end.

Output Format:

--------------

Print a space seperated list.

Sample Input-1:

---------------

20 25

Sample Output-1:

----------------

22 24

Explanation:

-----------

20 has 0 as digit so it's not self - supportive.

21 is not divisible by 2. so it's not self - supportive.

22 is divisible 2. so it's self - supportive.

23 is not divisible by both the digits 2 and 3. so it's not self - supportive.

24 is divisible by both 2 and 4. so it is self - supportive.

So 22 and 24 are self-supportive.

Sample Input-2:

---------------

50 80

Sample Output-2:

55 66 77 \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int start=sc.nextInt();

int end=sc.nextInt();

for(int i=start;i<=end;i++){

if(check(i)){

System.out.println(i);

}

}

}

public static boolean check(int n){

int x=n;

while(n>0){

int remainder=n%10;

if(remainder==0 || x%remainder>0){

return false;

}

n=n/10;

}

return true;

}

}

/\*Mr Kamalesh is working with Numbers.

He is given an number N. He wants to check whether N,

obeys following conditions or not:

- Initialize P=0, extract each digit 'd' from N, add them to P=> P = P + d^2.

- Add reset N value to P as N=P.

- and repeat the process until the N equals 1 or

if N loops endlessly in a cycle without having 1.

Your task is to check whether N becomes 1 or Not.

If yes, print "true", otherwise print "false".

Note: A^B indicates A to power of B.

Input Format:

-------------

An integer N

Output Format:

--------------

Print a boolean value.

Sample Input-1:

---------------

19

Sample Output-1:

----------------

true

Explanation:

------------

1^2 + 9^2 = 82

8^2 + 2^2 = 68

6^2 + 8^2 = 100

1^2 + 0^2 + 0^2 = 1

Sample Input-2:

---------------

20

Sample Output-2:

----------------

false \*/

import java.util.\*;

class Test{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int result=0;

while(n>0){

int remainder=n%10;

n=n/10;

result=remainder^2+n^2;

}

if(result==1){

System.out.println("true");}

else{

System.out.println("false");

}

}

}

Prefs solution:

import java.util.\*;

class d3p1{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

int n;

n=sc.nextInt();

Set<Integer> is=new HashSet<Integer>();

is.add(n);

while(true){

String x=n+"";

int y=0;

for(char c:x.toCharArray()){

y += (c-'0')\*(c-'0');

}

if(y==1){

System.out.println("true");

break;

}

else if(is.contains(y)){

System.out.println("false");

break;

}

else{

is.add(y);

n=y;

}

}

/\*Bob is working with numbers

He is given a list of numbers, his task is to find the priority number.

The priority number is the number that occured more than 'list.length/2' times.

You may assume that the list is non-empty

and the priority number always exist in the list.

Input Format:

-------------

line-1: an integer N, list size.

line-2: N space separated integers, numbers in the list[].

Output Format:

--------------

An integer, the priority number of the set.

Sample Input-1:

---------------

3

3 2 3

Sample Output-1:

----------------

3

Sample Input-2:

---------------

7

2 2 1 1 1 2 2

Sample Output-2:

----------------

2

\*/

import java.util.\*;

class Test

{

public static void main (String args[])

{

Scanner sc = new Scanner (System.in);

int n = sc.nextInt ();

int arr[] = new int[n];

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

{

arr[i] = sc.nextInt ();

}

int pri\_ele=arr[0];

int count=1;

int no\_pri=0;

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

if(arr[i]!=pri\_ele){

count--;

if(count==0){

no\_pri=1;

}

else if(count<0){

pri\_ele=arr[i];

count=1;

}

}

else{

count++;

}

}

System.out.println(pri\_ele);

}

}