Aditya Degree Colleges

Java Online Training Coding Test_6 Key

Date: 28-04-2020

Program – 1:

Let's play a game!! Given a matrix A[][] with N x M elements. Your task is to check that matrix is Super Similar or not. To perform this task you have to follow these rules:

Firstly all even index rows to be Rotated left and odd index rows to right, And Rotation is done X times(Index starting from zero). Secondly, after all the Rotations check if the initial and the final Matrix are same Print 1 else 0.

For Example:
A[][]={{1,2}, {5,6}};
X=1

Output: 0
Modified Matrix should look like:
A[][]={{2,1}, {6,5}};

Input:
The first line of input contains a single integer T denoting the number of test cases.

Then T test cases follow. Each test case consists of three lines.

First line of each test case consists of two space separated integers N and M, denoting the rows and columns respectively.

Second line of each test case consists of N*M space separated integers denoting the elements in the matrix in row major order.

Third line of each test case consists of an integer X, denoting the number of rotation.

Output:
It should be single line output, Print the respective output in the respective line.

Constraints:
1<=T<=100
1<=N,M<=10
1<=X<=10

Example:
Input:
2
2 2
1 2 5 6
1
2 4

1 2 1 2 2 1 2 1
2

Output:
0
1

Testcase1:
Input:
2
2 2
1 2 5 6
1
2 4
1 2 1 2 2 1 2 1
2
Output:
0
1
Testcase2:
Input:
2
3 3
1 1 1 1 1 1 1 1 1
3
4 4
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
4
Output:
1
1

Testcase3:
1
3 3
1 2 3 4 5 6 7 8 9
2
0

Source Code:

```
import java.util.*;
class Matrix_rotation
{
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		int m,n,ele1[][],ele2[][],ele3[][],i,j,r,testcases,f;
		testcases=sc.nextInt();
	while(testcases-->0)
	{
		f=1;
		m=sc.nextInt();
		n=sc.nextInt();
		ele1=new int[m][n];
		ele2=new int[m][n];
		ele3=new int[m][n];
		for(i=0;i<m;i++)
			for(j=0;j<n;j++)
			{
			    ele1[i][j]=sc.nextInt();
				ele3[i][j]=ele1[i][j];
			 }
		r=sc.nextInt();

		for(int k=0;k<m;k++)
		  if (k%2==0)
		      rotate_left(ele1[k],ele2,k,r);
		  else
			rotate_right(ele1[k],ele2,k,r);

		for(i=0;i<m;i++)
			{
				for(j=0;j<n;j++)
				 if(ele3[i][j]!=ele2[i][j])
					{
						f=0;
						break;
					 }
				if(f==0)
				  break;
			}
```

```
                System.out.print(f);
            }
        }
        public static void rotate_right(int x[],int ele2[][], int row,int r)
        {
            while(r>0)
            {
                int temp=x[x.length-1];
                for(int i=x.length-1;i>0;i--)
                {
                    ele2[row][i]=x[i-1];
                  x[i]=x[i-1];
                }
                 ele2[row][0]=temp;
                 x[0]=temp;
                r--;
            }
        }
        public static void rotate_left(int x[],int ele2[][],int row,int r)
        {
            while(r>0)
            {
                int temp=x[0];
                for(int i=0;i<x.length-1;i++)
                {  ele2[row][i]=x[i+1];
                   x[i]=x[i+1];
                }
                ele2[row][x.length-1]=temp;
          x[x.length-1]=temp;
              r--;
            }
         }
}
```

Program – 2:

Akaash was given an assignment by his teacher. He is given a integer N and another integer K. It is told that N is the integral part of K_{th} root of some integer. He needs to find out how many such integers are possible (Refer sample for explanation). Since he is weak at mathematics, help him do his assignment.

Input:
First line of input contains a single integer T denoting the number of test cases. The only line of each test case contains 2 space-separated integers N and K.

Output:
For each test case, print the required answer.

Constraints:
1 <= T <= 500
1 <= N <= 100
2 <= K <= 8

Example:
Input :
3
3 2
2 3
2 2
Output :
7
19
5

Explanation:
Case 1 :
3 can be the integral part of square root of following numbers:
9 10 11 12 13 14 15
Case 2 :
2 can be the integral part of cube root of following numbers:
8 9 10 11 ... 26
Case 3 :
2 can be integral part of square root of following numbers:
4 5 6 7 8

Testcase1:

Input:

3
3 2
2 3
2 2

Output :

7
19
5

Testcase2:

Input:

5
2 4
3 4
5 4
6 4
7 9

Output:

65
175
671
1105
93864121

Testcase3:

Input:

3
7 8
6 3
3 6

Output:

11012415
127
3367

Testcase4:

Input:

1
1 2

Output:

3

Source Code:

```
import java.util.*;
class NumberIntegral {
        public static void main (String[] args)
        {
                Scanner sc=new Scanner(System.in);
                int t=sc.nextInt();
                while(t-->0){
                   long n=sc.nextLong();
                   long n1=n+1;
                int n2=0;
                int m=sc.nextInt();
                int count=0;
                n=(long)Math.pow(n,m);
                n1=(long)Math.pow(n1,m);
                   if((n1-n)>0)
                        System.out.println(n1-n);
                   else
                        System.out.println(-(n1-n));

                }
        }
}
```

MCQ_Day6_KEY

Order of execution of constructors in Java Inheritance is
A) Base to derived class
B) Derived to base class
C) Random order
D) none
Answer: A

The first implicit statement added to the default constructor of the subclass is
A) super();
B) extend
C) super.className();
D) super.child();
Answer: A

Inheritance relationship in Java language is
A) Association
B) Is-A
C) Has-A
D) None
Answer: B

Advantage of inheritance in java programming is/are
A) Code Re-usability
B) Class Extendibility
C) Save development time
D) All
Answer: D

Predict the output

```
class X
{
   void method(int a)
   {
      System.out.println("ONE");
   }

   void method(double d)
   {
      System.out.println("TWO");
   }
```

```
}

class Y extends X
{
    @Override
    void method(double d)
    {
        System.out.println("THREE");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
          Y o=new Y();
          o.method(100);
    }
}
```

A)THREE
B)TWO
C)ONE
D)ALL
Answer:C

```
class A
{
    int i = 10;
 }
 class B extends A
{
    int i = 20;

}
public class Errors{
        public static void main(String[] args) {
         A a = new B();
        System.out.println(a.i);

    }
```

A) 20
B) 10
C) 0
D) Null
Answer: B

What will be the output of the following program?

```
class A
{
   static void methodOne()
   {
      System.out.println("AAA");
   }
}

class B extends A
{
   static void methodOne()
   {
      System.out.println("BBB");
   }
}

public class MainClass
{
   public static void main(String[] args)
   {
      A a = new B();
      a.methodOne();
   }
}
```

A) AAA
B) BBB
C) AAA BBB
D) BBB AAA
Answer: A

In the below example, is “methodOfX()” correctly overridden in the sub classes of Class X?

```
class X
{
   void methodOfX()
   {
      System.out.println("Class X");
   }
}

class Y extends X
{
   @Override
   private void methodOfX()
   {
      System.out.println("Class Y");
   }
}

class Z extends Y
{
   @Override
   public void methodOfX()
   {
      System.out.println("Class Z");
   }
}
```

A) True
B) False
C) None of the above
D) It is legal to create
Answer: B

Which of the following Line number is Invalid syntax?

```
Class Animal
{
}
Class Cat extends Animal
{
}
Line 1:Animal a=new Cat();
Line 2:Cat c=new Animal();
```

Line 3:Cat co=new Cat();
Line 4:Animal AO=new Animal();
A) Line 1
B) Line 2
C) Line 3
D) Line 4
Answer: B

Dynamic method dispatching is for the achieving of
A) Method Overloading
B) Method Overriding
C) Variable Overriding
D) Both B and C
Answer:B

Which of the following is not a wrapper Class
A) Integer
B) String
C) Character
D) Boolean
Answer: B

Multiple Inheritance in java is achieved by
A) Multi Threading
B) Garbage Collection
C) Exception
D) Interface
Answer:D

Super keyword is invoked from the
A) Super class
B) Sub Class
C) Main Class
D) Independent Class
Answer: B

Predict the out put

```
  Scanner s=new Scanner(System.in);
        char ch;
    ch=s.next().charAt(1);
    System.out.print(ch);
```

if the input is world then

A) w

B) o

C) r

D) d

Answer:B

```
class Test
{
  public static void main(String args[])
  {
    Test obj=new Test();
    obj.start();
  }
  void start()
   {
     String str1="do";
     String str2=method(str1);
     System.out.print(": "+stra+strb);
    }
   String method(String stra)
   {
       stra=stra+"good";
       System.out.print(stra);
       return "good";
   }
}
```

A) dogood: dogood

B) dogood: dodogood

C) dogood: dogoodgood

D) dogood: gooddogood

ANSWER:A

Wrapper classes are used for the converting
A) Primitive data types into Objects
B) Converting String Class into Object
C) Converting Throwable Class inot Object
D) Converting Thread Class into Object
Answer:A

What is the output of the following program?

```
class Derived
{
   public void getDetails()
   {
      System.out.printf("Derived class ");
   }
}
public class Test extends Derived
{
   public void getDetails()
   {
      System.out.printf("Test class ");
      super.getDetails();
   }
   public static void main(String[] args)
   {
      Derived obj = new Test();
      obj.getDetails();
   }
}
```

A) Test class Derived class
B) Derived class Test class
C) Compilation error
D) Runtime error
Answer: A

Prevention of Inheritance in java is possible
A) this
B) final
C) finalize
D) finally
Answer:B

Which of the following are incorrect form of StringBuffer class constructor?
A) StringBuffer()
B) StringBuffer(int size)
C) StringBuffer(String str)
D) StringBuffer(int size , String str)
Answer: D

Which of these methods of class StringBuffer is used to extract a substring from a String object?
A) substring(int start,int end)
B) Substring(int start)
C) SubString(int start,int end)
D) None of the mentioned
Answer: A