DO NOT WRITE ANYTHING ON QUESTION PAPER EXCEPT ROLL NO.

CHITKARA UNIVERSITY

2022-2023

Sessional Test I – April, 2023

Roll No: …… [Total No. of Pages: 8]

Programme: B.E. (CSE) **Time: 90 minutes**

Course Title: Core Java

Course Code: CS109 **Max. Marks: 40**

General Instructions:

- Follow the instructions given in each section.

Section – A

(Q 1 to 10: Each Question carries 1 mark)

Q 1. Which of these is necessary to specify at time of array initialization?

a. Row
b. Column
c. Both Row and Column
d. None of the mentioned

Q 2. The order of calling constructor in case of inheritance is

a. Super class constructor and then Subclass Constructor
b. Subclass constructor and then Super class Constructor
c. Super class and subclass constructors are independent
d. Super class and subclass constructors can be called in any order

Q 3. Which of these statements are incorrect?

a. Equal to operator has least precedence
b. Brackets () have highest precedence
c. Division operator, /, has higher precedence than multiplication operator
d. Addition operator, +, and subtraction operator have equal precedence

Q 4. When does method overloading is determined?

a. At run time
b. At compile time
c. At coding time
d. At execution time

Q 5. What will be the output of the following program?

```
public class Test {
      static int number = 0 ;
public static void main(String[] args) {
      number++;
      Test.number++;
   System.out.println(number);
 }
}
```

a. 0
b. 2
c. 1
d. Error : Statement Test.number++ is not correct.

Q 6. Select the valid declaration and initialization of an array in JAVA.

a. int[] A = {}
b. int[] A = {1, 2, 3}

c. int[] A = (1, 2, 3)
d. int[][] A = {1,2,3}

Q 7. Can command line arguments be converted into int automatically if required?

a. Yes
b. No
c. Compiler Dependent
d. Only ASCII characters can be converted

Q 8. Find the output of given program:

```
class Arraymcq
{
public static void main(String args[])
{
 int[] intArray = new int[]{ 1,2,3,4,5,6,7,8,9,10 };
System.out.println(intArray[0]);
}
}
```

a. 1
b. [1,2,3]
c. Error : Array Index out of Bounds Exception
d. 1 2 3

Q 9. How many objects are collected by garbage collector in the following code:

```
public class Test
{
	public static void main(String[] args) {
		Test t1 = new Test();
		Test t2 =new Test();
		Test t3 = new Test();
		t1 = t2 = t3;
		System.gc();
	}
}
```

a. 0
b. 1
c. 2
d. 3

Q 10. What will be the output of the following program?

```
public class Test {
	public static void main(String[] args) {
		Base s = new Derived();
		s.fun();
	}
}
class Base {
	public void fun() {
		System.out.println("Base");
	}
}
class Derived extends Base {
	public void fun() {
		System.out.println("Derived");
```

```
    }
}
```

a. Base
b. Base can not override the method “fun()”
c. Derived
d. The statement “Base s = new Derived ();” is illegal.

Section - B
(Q 11 to 15 : Each question carries 2 marks)

Q 11. Mention the correct output of the following Java program. Where the command line ex ecution is done as – “java Output This is a command Line”?

```
class Output
{
public static void main(String args[])
 {
      System.out.print(args[0]);
 }
}
```

a. java
b. Output
c. This
d. is

Q 12. What will be the output of the following code? class work

```
{
public static void main(String args[])
{
int x = 1;
x= x<<1>>2;
System.out.print(x);
}
}
```

a. 0
b. 1
c. 2
d. 3

Q 13. Identify what can directly access and change the value of the variable res.

```
Public class Solution{
    Private int res = 100;
}
```

Choices *(With Correct Answers)*

a. Any class
b. Only Solution class
c. Any class that extends Solution
d. None of the these

Q 14. What is the output of the below Java program?

```
public class MethodOverloading2
{
 int info()
 {
```

```
  System.out.println("Welcome");
  return 0;
 }

 void info()
 {
  System.out.println("Congratulations");
 }

 public static void main(String[] args)
 {
  MethodOverloading2 m = new MethodOverloading2();
  int a = m.info();
 }
}
```

a. Congratulations
b. Welcome
c. Compiler error
d. None

Q 15. What will be the output of the program?

```
public class Test {
  public static void main(String args[]) {
        int arr[] = new int[2];
        System.out.println(arr[0]);
        System.out.println(arr[1]);
  }
}
```

a. 0 0
b. garbage value garbage value
c. Compiler Error
d. Exception

Section - C
(Q 16 to 17: Each Q carries 5 marks)

Q16. Given three numbers x, y and z of which aim is to get the largest among these three numbers and print the result.

Input format:
First line of the input contains 3 space separated numbers entered by the user.

Output format:
On a single line of output print the largest number
0 <= x,y,z <= 10000000

Sample Input:
5 10 3
Sample Output:
10 is the largest number
Sample Input:
5 5 5
Sample Output:
5 is the largest number

Test Cases:

Input	Output
0 1 2	2 is the largest number
10 1000 100	1000 is the largest number
500 50 50	500 is the largest number
56 55 54	56 is the largest number
100 100 100	100 is the largest number

Solution:

```
import java.util.*;
class Largest {
        static int biggestOfThree(int x, int y, int z)
        {
        if (x >= y && x >= z)
        return x;
        else if (y >= x && y >= z)
        return y;
        else
        return z;
        }
        public static void main(String[] args)
        {
                Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        int b = sc.nextInt();
        int c = sc.nextInt();
        int largest;
        largest = biggestOfThree(a, b, c);
        System.out.println(largest + " is the largest number");
        sc.close();
        }
}
```

Q17: Given an n x m matrix, where every row and column is sorted in increasing order, and a number x . Find if element x is present in the matrix or not.

Input Format

First line consists of two space separated integers N and M, denoting the number of element in a row and column 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 contains a single integer x, the element to be searched.

Constraints

1 <= N,M <= 30 0 <= A[i] <= 100

Output Format

Print 1 if the element is present in the matrix, else 0.

Sample Input
3 3
3 30 38
44 52 54
57 60 69
62
Sample Output
0
Explanation
Search the element in the sorted matrix. If the element is present print 1 otherwise print 0. In the sample input,in first case 62 is not present in the matrix so 0 is printed. Similarly, for second case 55 is present in the matrix so 1 is printed.
Test Cases:

Input	Output
3 3 3 30 38 44 52 54 57 60 69 62	0
3 3 3 4 5 6 7 9 10 11 12 7	1
3 4 13 14 15 16 23 24 25 26 33 34 35 36 25	1

SOLUTION:

```
import java.util.*;
public class Main {
        public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
                        int row = sc.nextInt();
                        int col = sc.nextInt();
                        int[][] arr = new int[row][col];
                        for (int i = 0; i < row; i++) {
                                for (int j = 0; j < col; j++) {
                                        arr[i][j] = sc.nextInt();
                                }
                        }
                        int item = sc.nextInt();
                        Print(arr, item);

        }
```

```
public static void Print(int[][] arr, int item) {
        int i = arr.length - 1;
        int j = 0;

        while (i >= 0 && j <= arr[0].length - 1) {
            if (item == arr[i][j]) {
                System.out.println("1");
                return;
             }
             else if (item > arr[i][j]) {
                j++;
             } else  {
                i--;
             }
        }
System.out.println("0");
        return;
}}
```

Section - D

(Q 18: Q carries 10 marks)

Q18: Andrew manages a pipe warehouse. He wishes to automate the process of transferring the pipes from the warehouse to the carrier truck. There are N pipes with a length of 1 to 9 units placed in the warehouse vertically along a wall. In the automated system, a drone picks the pipes by length and carries them to the carrier truck. In each turn, the drone moves from left to right to find the pipe, whose length is greater than the pipe on its left. After finding the pipe, the drone takes the pipe to the carrier truck. The drone repeats this process until it has no more pipes to pick or no pipe meets the above-said criteria. Write a Java program to output the length of pipes, which will remain in the warehouse after the drone has completed this process.

Input Format

The first line of the input consists of an integer - numOfPipes, representing the number of pipes in the warehouse (N).

The second line consists of N space-separated integers - len[0], len[1], ... len[N-1], representing the length of the pipes.

Output Format

Print integers representing the length of the remaining pipes in the warehouse. Don't print any space between integers

Sample Input 1

5

4 5 2 1 3

Sample Output 1

421

Explanation

In the first turn, the drone picks the pipe with length 5 as 5 > 4. So, the remaining pipes are 4, 2, 1, and 3.

In the next turn, the drone picks the pipe with length 3 as 3 > 1. So, the remaining pipes are 4, 2, and 1.

Sample Input 2
5
3 2 4 6 5
Sample Output 2
32
Explanation
In the first turn, the drone picks the pipe with length 4 as 4 > 2. So, the remaining pipes are 3, 2, 6, and 5.
In the next turn, the drone picks the pipe with length 6 as 6 > 2. So, the remaining pipes are 3, 2, and 5.
In the next turn, the drone picks the pipe with length 5 as 5 > 2. So, the remaining pipes are 3 and 2.

	Test Case 1	**Test Case 2**	**Test Case 3**	**Test Case 4**	**Test Case 5**
Input	5 1 2 3 4 5	5 3 5 4 8 7	7 1 5 2 6 3 7 4	9 9 8 7 6 5 4 3 1 2	6 3 9 2 8 1 7
Output	1	3	1	98765431	321

#Solution

```
import java.util.Scanner ;
class Main
{
	public static void main(String[] args)
	{
		Scanner scan = new Scanner(System.in) ;
		int n = scan.nextInt() ;
		int [] pipes = new int[n] ;
		for(int i = 0 ; i<n ; i++)
		{
			pipes[i] = scan.nextInt() ;
		}
		System.out.print(pipes[0]) ;
		int temp = pipes[0];
		for(int i = 1 ; i<n ; i++)
		{
			if(pipes[i]<temp)
			{
				System.out.print(pipes[i]);
				temp = pipes[i];
			}
		}
	}
}
```