CHITKARA UNIVERSITY

2022-2023

Sessional Test I – April, 2023

Roll No: …… [Total No. of Pages: 9]

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)

Q1. What is the output of the below Java code?

```
boolean[] ary = {true, false, true, true};
for(int i=0; i<ary.length; i++)
{
 System.out.print(ary[i] +",");
}
```

a. true,true,true,true,
b. true,false,false,true
c. true,false,true,true,
d. Compiler error

Q2. When an expression consists of int, double, long, float, then the entire expression will get promoted into a data type that is:

a. float
b. double
c. int
d. long

Q3. What is the value of "age" in the below Java program with a DO-WHILE loop?

```
int age=20;
do
{
  age++;
}while(age<20);
System.out.println(age);
```

a. 20
b. 21
c. Compiler error
d. None

Q4. What is the output of the below Java program?

```
String str[] = {"A","B","C"};
int i=0;
do
{
```

```
 if(i>= str.length)
break;
 System.out.print(str[i] + ",");
 i++;
}while(true);
```

a. A,B,C,
b. A,B,C
c. Runtime Exception with Index Of Bounds Exception
d. Compiler error

Q5. To pass a string as a command-line argument in Java, you need to surround the text within a pair of _.

a. Single Quotes ('abc def')
b. Double Quotes ("abc def")
c. Double Spaces(abc def)
d. Triple Single Quotes ('"abc def"')

Q6. Identify the infinite loop in java can be defined as.

a. for(; ;)
b. for(int i = 0; i < 1; i--)
c. for(int i = 0; ;i++)
d. All of the above

Q7. When does Overloading not occur?

a. When more than a single method has the same name, yet different types or number of parameters and different method signature.

b. When more than a single method has the same name, the same signature, but have different numbers of signature.

c. When more than a single method has the same signature, same name, and the same number of parameters have different types.

d. When more than a single method has the same name, the same number and types of parameters, and yet different signatures.

Q8. Which of this method is given parameter via command line arguments?

a. main()
b. recursive() method
c. Any method
d. System defined methods

Question 9. What do you mean by >>> operator in Java?

a. Left Shift Operator
b. Right Shift Operator
c. Zero Fill Right Shift
d. Zero Fill Left Shift

Q10. Out of these, which one is the correct way of calling a constructor that has no parameters of the superclass A by the subclass B?

a. superclass.();
b. super(void);
c. super();
d. super.A();

Section - B

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

Q11. What will be the output

```
public class MyFirst {
        public static void main(String[] args) {
          MyFirst obj = new MyFirst(n);
}
static int a = 10;
static int n;
int b = 5;
int c;
public MyFirst(int m) {
         System.out.println(a + ", " + b + ", " + c + ", " + n + ", " + m);
  }
// Instance Block
 {
       b = 30;
       n = 20;
 }
// Static Block
 static
{
           a = 60;
       }
}
```

a. 10, 5, 0, 20, 0
b. 10, 30, 20
c. 60, 5, 0, 20
d. 60, 30, 0, 20, 0

Q12. What is the output of the below Java program?

```
public class TestingConstructor
{
 void TestingConstructor()
 {
      System.out.println("Amsterdam");
 }
 TestingConstructor()
 {
      System.out.println("Antarctica");
 }
 public static void main(String[] args)
 {
   TestingConstructor tc = new TestingConstructor();
 }
```

```
}
```

a. Antarctica
b. Amsterdam
c. No output
d. Compiler error

Q13. What will be the output of the following program?

```
class B
{
      static int count = 100;
      public void increment()
      {
         count++;
      }
      public static void main(String []args)
      {
         B b1 = new B();
         b1.increment();
         B b2 = new B();
         System.out.println(b2.count);   // line 13
      }
}
```

a. 100
b. 101
c. Error in line 13
d. 0

Q14. What is the output of the below Java program with constructors?

```
public class Constructor2
{
 int count=10;
 Constructor2(int count)
 {
  System.out.println("Count=" + count);
 }
 public static void main(String[] args)
 {
     Constructor2 con = new Constructor2();  } }
```

a. Count=0
b. Count=10
c. Compiler error
d. None of these

Q15. What is the output

```
public class comma_operator
  {
     public static void main(String args[])
     {
         int sum = 0;
         for (int i = 0, j = 0; i < 5 & j < 5; ++i, j = i + 1)
            sum += i;
    System.out.println(sum);
     }
  }
```

a. 5
b. 6
c. 14
d. Compilation error

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

Q16. Given an array A of size N-1 such that it only contains distinct integers in the range of 1 to N. Find the missing element.

Input Format
The first and only line contains an integer N denoting the size of the array. The second line contains array elements A.

Output format
Find the missing element and print it.

Constraints
$1 \leq N \leq 10^6$
$1 \leq A[i] \leq 10^6$

Time Limit
1 second

Example
Sample Input
5
1 2 3 5
Sample Output
4
Sample test case explanation
The missing number in the array is 4.

Default Code

```
import java.util.*;
import java.io.*;
public class Main {
        public static void main(String[] args) throws IOException {
        }
}
```

SOLUTION

```
import java.util.Scanner;
public class Solution {
        public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int[] array = new int[n-1];
        for (int i = 0; i < n-1; i++) {
        array[i] = sc.nextInt();
        }
        sc.close();
        int missingNumber = MissingNumber(array, n);
        System.out.println(missingNumber);
        }
        static int MissingNumber(int array[], int n) {
```

```
        int i, total;
        total = (n + 1) * n / 2;
        for (i = 0; i < n - 1; i++)
        total -= array[i];
        return total;
        }
}
```

Test Cases:

Test case 1	Test case 2	Test case 3	Test case 4
Input 10 2 8 7 6 1 3 5 4 10	Input 5 1 2 3 5	Input 20 3 4 16 17 19 15 13 20 8 6 14 7 10 2 1 9 5 18 12	Input 70 67 51 15 48 45 21 8 38 14 63 6 42 2 65 25 39 58 4 17 70 64 55 30 43 29 37 18 1 49 10 22 9 19 36 56 7 28 53 24 32 41 52 62 44 35 61 69 54 13 11 47 59 50 66 26 34 31 5 12 40 23 68 20 60 3 27 33 57 16
Output 9	Output 4	Output 11	Output 46

Q17. Given an array and you need to find out the contiguous subarray that has the largest sum and return the sum of the subarray

Input Format

A number representing n length of array. N lines representing the n elements of the array. The last line takes A number representing the target sum.

Constraints

The length of the array must be less than 10000

Output Format

Print the sum of the maximum subarray

Sample Input

9
-2 1 3 4 -1 2 1 -5 4

Sample Output

6

Explanation

{-2,1,-3,4,-1,2,1,-5,4}
The resultant will be 6.
{4,-1,2,1}is the subarray that has the largest sum = 6.

SOLUTION:

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

```
    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<n;i++){
    arr[i]= sc.nextInt();
    }
   System.out.println (max_sum( arr));
    }
    public static int max_sum(int[] arr) {
                    int ans = Integer.MIN_VALUE;
    int sum = 0;
                    for (int i = 0; i < arr.length; i++) {
                                    sum += arr[i];
                                    ans = Math.max(ans, sum);
            if(sum<0)
            sum=0;
                                }
                    return ans;
    }
}
```

Test Cases:

Input	Output
5 -4 -1 0 3 10	13
4 2 5 -8 9	9
6 3 5 7 -3 -7 6	15
4 -1 -2 -3 -4	-1

Section - D

(Q 18: Question carries 10 marks)

Q18: Given an array A of N integers. Check whether it contains a triplet that sums up to zero.

Input Format

The first line contains an integer N denoting the size of the array. The second line contains elements of the array A.

Output format

Print 1 if it contains a triplet that sums up to zero else print 0.

Constraints

1 <= N <= 10^4

-10^6 <= A[i] <= 10^6

Example

Sample Input
5
0 -1 2 -3 1
Sample Output
1
Sample test case explanation
0, -1 and 1 forms a triplet with sum equal to 0.

SOLUTION

```
import java.util.*;
import java.io.*;
public class Main
{
        public static class Solution
        {
      public boolean findTriplets(int arr[],int n)
      {
      Arrays.sort(arr);
      for(int i=0;i<n-2;i++)
         {
         int l=i+1;
         int r=n-1;
         int x=arr[i];
         while(l<r)
         {
            if(x+arr[l]+arr[r]==0)
                  return true;
            else if(x + arr[l] + arr[r] < 0)
                  l++;
                  else
                  r--;
         }
            }
      return false;
         }
         }
         public static void main(String[] args)
         {
      Scanner input = new Scanner(System.in);
      int n = input.nextInt();
      int[] arr = new int[n];
      for (int i = 0; i < n; i++) {
         arr[i] = input.nextInt();
         }
      Solution solution = new Solution();
         if (solution.findTriplets(arr, n)) {
         System.out.println("1");
         } else {
         System.out.println("0");
         }
```

```
        }
}
```

Test Cases:

Test case 1	Test case 2	Test case 3	Test case 5	Test case 4
Input 10 66 87 99 85 50 93 98 84 14 47	Input 50 -34 -15 11 78 -7 -2 -16 -86 -53 -56 -17 -12 -80 99 -93 -63 41 2 90 33 -59 -99 -36 -76 29 66 -80 100 65 -43 29 -29 31 94 -84 -99 97 -17 -28 91 2 98 -46 57 25 15 -86 -86 31 -22	Input 70 66 87 99 85 50 93 98 84 14 47 44 83 88 20 71 7 37 13 98 62 5 41 1 64 24 1 38 20 72 37 57 1 71 3 66 16 1 69 83 72 63 70 54 29 14 14 3 78 79 20 42 24 95 76 50 28 36 17 41 83 0 48 3 69 9 50 21 63 86 8	Input 8 1 2 3 4 -2 7 8 10	Input 100 -34 -15 11 78 -7 -2 -16 -86 -53 -56 -17 -12 -80 99 -93 -63 41 2 90 33 -59 -99 -36 -76 29 66 -80 100 65 -43 29 -29 31 94 -84 -99 97 -17 -28 91 2 98 -46 57 25 15 -86 -86 31 -22 -80 18 70 -76 -5 78 -72 64 -83 -59 -17 -100 -52 18 26 -97 -31 -91 2 -50 9 -79 91 -14 36 36 46 7 17 -56 85 -46 -62 65 -65 5 -34 -45 -100 -26 -34 34 90 55 46 -91 80 76 -100 -77
Output 0	Output 1	Output 0	Output 0	Output 1