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)

Q1: Java array is a collection of __.

a. similar type of elements
b. different type of element
c. heterogeneous data
d. Both A and C

Q2: Which operator is used to check object-type at runtime?

a. ternary operator
b. Instance of operator
c. type operator
d. length operator

Q3: How many of the following are legal declarations?

```
[]double lion;
double[] tiger;
double bear[];
```

a. None
b. One
c. Two
d. Three

Q4: Which statement does not create an object of class Student{} in Java ?

a. new Student();
b. Student s1 = new Student(), s2 = new Student();
c. Student s1;
d. Student ss = new Student();

Q5. What will be the return type of a method that does not return any value.

a. int
b. void
c. double
d. None

Q6. Identify the infinite loop in java.

a. for(; ;)
b. for(int i = 0; i < 1; i--)
c. for(int i = 0; ;i++)

d. All of the above

Q7. The system stores parameters and local variables in **__**whenever a method is invoked.

a. Heap
b. Stack
c. Array
d. Tree

Q8. What will be the output of the following program?

```
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

Q9. What will be the output of the following program?

```
public class Main
{
public static void main(String args[])
{    int arr[] = {10, 20, 30, 40, 50};
     for(int i=0; i < arr.length; i++)
      {
            System.out.print(" " + arr[i]);
       }
 }
}
```

a. 10 20 30 40 50
b. Compile Time Error
c. Runtime Error
d. 10 20 30 40

Q10. 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

Section - B

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

Q11. What will be the output of the following java code?

```
class Animal
{
public void move()
{
System.out.println("Animals can move");
}
}
class Dog extends Animal
{
public void move()
{
System.out.println("Dogs can walk and run");
}
public void bark()
{
System.out.println("Dogs can bark");
}
}
public class TestDog {public static void main(String args[])
{
Animal b = new Dog();
b.move();
}
}
```

a. Dogs can bark
b. Dogs can walk and run
c. Runtime Exception
d. Animal can walk

Q12. What will be the output?

```
public class Test {
static int start = 1;
final int end;
public Test(int x)
{
x = 2;
end = x;
}
public void fly(int distance)
{
System.out.println(end-start);
System.out.println(distance);
}
public static void main(String []args)
{
```

```
new Test(15).fly(5); }
}
```

a. 5 15

b. 1 5

c. compliation error

d. runtime error

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

```
public class solution
{
public static void main(String args[])
{
int y, x = 3;
y = --x + x++ - --x * --x * x;
System.out.println(y + " " + x);
}
}
```

a. 2 2

b. 2 1

c. 1 1

d. 1 2

Q14. How many columns does a have if it is created as follows:

```
int[][] a = { {1, 4, 7, 8}, {3, 2, 3, 4}};
```

a. 2

b. 4

c. 8

d. None of the these

Q15. What will be the output of the following code?

```
public class calculation
{
        public static void main(String args[])
        {
                int a = 2, b = 2, c = 3;
                if (--a == b && --b == c );
                else if (a == b & --b == c);
                System.out.print(a + " " + b + " " + c);
        }
}
```

a. 1 2 3

b. 2 2 3

c. 1 1 3

d. 1 0 3

Section - C

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

Q16. Write a program in Java to convert a given decimal number to binary form and print the resultant number.

Input format:

First line of the input contains a single decimal input number entered by the user.

0 <= n <= 10000000

Output format:

On a single line of output print the binary number as a String

Sample Input:

7

Sample Output:

111

Sample Input:

10

Sample Output:

1010

Test Cases:-

Input	**Output**
33	100001
100000000	101111101011110000100000000
106	1101010
0	0
1	1

Solution:

```
import java.util.*;
public class DecimalToBinary {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int decimalNumber = sc.nextInt();
      String binaryNumber = "";
      if(decimalNumber == 0){
           binaryNumber = "0";
      }
      Else {
         while(decimalNumber > 0){
            binaryNumber = (decimalNumber % 2) + binaryNumber;
            decimalNumber = decimalNumber / 2;
         }
      }
      System.out.println(binaryNumber);
      sc.close();
   }
}
```

Q17. Write a program in java to print Armstrong number between two input numbers. An Armstrong number is a number (with digits n) such that the sum of its digits raised to nth power is equal to the number itself.

Input format:

First line of the input contains two input numbers entered by the user.

Output format:

On a single line of output print the Armstrong numbers separated by space

n1 < n < n2
1 < n < 10000

Sample Input:
5 20

Sample Output:
6 7 8 9

Sample Input:
50 400

Sample Output:
153 370 371

Test Cases:

Input	Output
5 7	6
60 160	153
1000 10000	1634 8208 9474
100 500	153 370 371 407
1 10	2 3 4 5 6 7 8 9

Solution:

```
import java.math.*;
import java.util.*;
 class Armstrong {
        static void ArmstrongNum(int l, int h)
        {
        for (int j = l + 1; j < h; ++j) {

        int y = j;
        int N = 0;
        while (y != 0) {
                y /= 10;
                ++N;
        }

        int sum_power = 0;
        y = j;
        while (y != 0) {
                int d = y % 10;
                sum_power += Math.pow(d, N);
                y /= 10;
        }
        if (sum_power == j)
                System.out.print(j + " ");
        }
        }
        public static void main(String args[])
        {
```

```
        Scanner sc = new Scanner(System.in);
        int n1 = sc.nextInt();
        int n2 = sc.nextInt();
        ArmstrongNum(n1, n2);
        System.out.println();
        sc.close();
}
}
```

Section - D

(Q 18: Question carries 10 marks)

Q18: In a gaming hub, N number of players were playing the same type of game. All players got stuck at the pillar level in the game, each with a different score. The owner of the gaming hub announced that players can pass that level if they can break two pillars. Both pillars have their own health points. The trick is to break the pillar at a time if a score equal to the pillar's health is obtained by multiplying the player's current score with any number. The same trick is to be used for both pillars. If no number can be multiplied by the player's score to make the score equal to the pillar's health, then that player loses. If the player is not able to break both pillars, then he will not be able to clear the level. Write a Java program to find the total number of players who will clear that level of the game.

Input Format:

The first line of the input consists of an integer -num Players representing the number of players (N).

The second lines consist of N space-separated integers - score1, score2, ..., scoreN, representing the score of each player. The last line of the input consists of two integers - health1, health2 representing the health of both the pillars respectively.

Output Format

Print the count of the players who will clear that level of the game

Sample Input 1

```
5
15 5 3 7 9
90 30
```

Sample Output 1

```
3
```

Explanation

Scores 15, 5 and 3 can be multiplied by another number to get 90 and 30 which will break the pillar. So, the output is 3.

Sample Input 2

```
5
15 5 3 7 9
135 90
```

Sample Output 2

```
4
```

Explanation

Scores 15, 5, 3, 9 can be multiplied by another number to get 135 and 90 which will break the pillar. So, the output is 4.

	Test Case 1	**Test Case 2**	**Test Case 3**	**Test Case 4**	**Test Case 5**

Input	6 15 5 3 7 9 45 135 90	8 15 5 3 7 9 45 10 20 15 90	7 10 20 30 40 50 60 70 20 10	8 11 15 19 27 29 34 19 32 23 25	5 15 5 3 7 9 135 90
Output	5	3	1	0	4

#Solution

```
import java.util.*;
class Main
{
        public static int fun(int n, int arr[], int x, int y)
        {
                int i, count = 0;
                for(i = 0; i < n; i++)
                {
                        if(x % arr[i] == 0 && y % arr[i] == 0)
                        count++;
                }
                return count;
        }
        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();
                }
                int x = sc.nextInt();
                int y = sc.nextInt();
                System.out.print(fun(n, arr, x, y));
        }
}
```