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: 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 will be the output of the following program?

```
public class Test {
 public static void main(String[] args) {
    String s = new String("5");
    System.out.println(1 + 10 + s + 1 + 10);
 }
}
```

a. 11511
b. 1105110
c. 115110
d. 27

Q2. What will be the output of the following program?

```
public class LoopExample {
 public static void main(String[] args) {
  for(int i=0; i<10; i+=2){
   System.out.print(i);
  }
 }
}
```

a. 0123456789
b. 02468
c. 0246810
d. 123456789

Q3. What is the output of the following code?

```
int[] arr = new int[5];
System.out.println(arr[3]);
```

a. 0
b. null
c. Array Index Out of Bounds Exception
d. none of the above

Q4. When is the object created with new keyword?

a. At compile time
b. Depends upon the code

c. At run time
d. None of the above

Q5. What is the output of the following code?

```
String[] arr = {"apple", "banana", "cherry"};
System.out.println(arr[1]);
```

a. apple
b. banana
c. cherry
d. Array Index Out Of Bounds Exception

Q6. When an array is passed to a method, what does the method receive?

a. The reference of the array
b. A copy of the array
c. Length of the array
d. Copy of the first element

Q7. 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

Q8. Which of these cannot be used for a variable name in Java?

a. identifier & keyword
b. identifier
c. keyword
d. none of the mentioned

Q9. What is the output of the below Java program with WHILE, BREAK and CONTINUE?

```
public class Test {
 public static void main(String[] args) {

int cnt=0;
while(true)
{
  if(cnt > 4)
 break;
  if(cnt==0)
  {
 cnt++;
 continue;
  }
  System.out.print(cnt + ",");
  cnt++;
}

 }
}
```

a. 0,1,2,3,4,

b. 1,2,3,4,
c. 1,2,3,4
d. Compiler error

Q10. What will be the output of the following program?

```
class jump_statments
{
   public static void main(String args[])
   {
       int x = 2;
       int y = 0;
       for ( ; y < 10; ++y)
       {
          if (y % x == 0)
             continue;
          else if (y == 8)
              break;
          else
            System.out.print(y + " ");
       }
   }
}
```

a. 1 3 5 7
b. 2 4 6 8
c. 1 3 5 7 9
d. 1 2 3 4 5 6 7 8 9

Section - B

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

Q11. How many objects are eligible for garbage collection after execution of line ?

```
public class Test
{
        public static void main(String[] args)
        {
                m1(); // Line
        }
        static void m1()
        {
                Test t1 = new Test();
                Test t2 = new Test();
        }
}
```

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

Q12. What is the output of the following code fragments?

```
int []fun = new int [5];

fun[0] = 1;
```

```
fun[1] = 2;

fun[2] = 3;

fun[3] = 4;

fun[4] = 5;

int j = 3;

System.out.println(fun[ j-1]) ;
```

a. 1
b. 2
c. **3**
d. 4

Q13. Find out the invalid for loop declaration?

a. for (int i = 7; i <= 77; i += 7)
b. for (int i = 99; i >= 0; i / 9)
c. for (int i = 20; i >= 2; - -i)
d. for (int i = 2; i <= 20; i = 2* i)

Q14. What will be the output of following code?

```
public class Test
{
   public static void main (String[] args)
   {
      int arr1[] = {1, 2, 3};
      int arr2[] = {1, 2, 3};
      if (arr1 == arr2)
         System.out.println("Same");
      else
         System.out.println("Not Same");
   }
}
```

a. Compilation Error
b. Runtime Error
c. Same
d. Not Same

Q15. Find out the output of following code

```
public class Main
 { public static void main(String args[])
{ int i;
for(i = 2; i < 6; i++)
{
if(i > 3)
continue;
 }
System.out.println(i);
 } }
```

a. **6**
b. 7
c. 8
d. 9

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

Q16. Imagine you are a software developer working on a program that checks the divisibility of a positive integer n, given by the user. The program should follow the rules below:
If the integer is divisible by 2, print "Two".
If it is divisible by 3, print "Three".
If it is divisible by 11, print "Eleven".
If it is not divisible by any of the above three rules, print "-1".
Note: If N is divisible by more than one of the above given numbers, print the one which is largest.

Input format:
First line contains an integer N.

Output format:
Output the string as required.

Sample Input:
3

Sample Output:
Three

Sample Input:
6

Sample Output:
Three

	Test Case 1	Test Case 2	Test Case 3
Input	22	4	23
Output	Eleven	Two	-1

Solution:

```
import java.util.*;
import java.io.*;
public class Main{
    public static void main (String[] args) {
      Scanner sc = new Scanner(System.in);
         int n =sc.nextInt();
         Geeks obj=new Geeks();
         obj.isDivisibleByPrime(n);
   }
}
class Geeks {
    static void isDivisibleByPrime (int n)
    {
      if((n%11) ==0)
      System.out.println("Eleven");
      else if((n%3) ==0)
      {
          System.out.println("Three");
      }
```

```
        else if((n%2) ==0)
        {
            System.out.println("Two");
        }
        else
        {
            System.out.println("-1");
        }
    }
}
```

Q17. Given two arrays A and B of size N and M respectively. The task is to find the number of elements in the union between these two arrays.
Note - Union of the two arrays can be defined as the set containing distinct elements from both the arrays. If there are repetitions, then only one occurrence of the element should be printed in the union.

Input Format

The first line contains two integers N and M denoting the size of two arrays A and B.
The second line contains the array elements of A as A_1,A_2,...,A_N-1.
The third line contains the array elements of B as B_1,B_2,...B_N-1.

Output format

Print the number of elements in the union between these two arrays A and B.

Constraints

1<=N,M<=100
0<=A_i,B_i<=100

Time Limit

1 second

Example

Sample Input

5 3
1 2 3 4 5
1 2 3

Sample Output

5

Sample test case explanation

1, 2, 3, 4 and 5 are the elements which come in the union set of both arrays. So the count is 5.

Solution:

```
import java.util.*;
import java.io.*;
import java.util.Scanner;
public class Main {
      public static void main(String[] args) {
       Scanner scanner = new Scanner(System.in);
       int n = scanner.nextInt();
       int m = scanner.nextInt();
       int[] a = new int[n];
       for (int i = 0; i < n; i++) {
          a[i] = scanner.nextInt();
       }
       int[] b = new int[m];
```

```
        for (int i = 0; i < m; i++) {
            b[i] = scanner.nextInt();
        }
        int result = doUnion(a, n, b, m);
        System.out.println(result);
        scanner.close();
    }
    public static int doUnion(int a[], int n, int b[], int m) {
        HashSet<Integer> s = new HashSet<>();
        for (int i = 0; i < n; i++)
            s.add(a[i]);
        for (int i = 0; i < m; i++)
            s.add(b[i]);
        return s.size();
    }
}
```

Test Cases:

Test case 1	Test case 2	Test case 3	Test case 4
Input 10 4 66 87 99 85 50 93 98 84 14 47 44 83 88 20	Input 50 50 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 8 18 81 44 57 90 54 88 38 88 37 35 66 100 55 77 0 74 66 6 62 27 18 91 9 52 78 48 0 23	Input 20 30 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	Input 10 50 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
Output 14	Output 61	Output 36	Output 42

Section - D

(Q 18: Question carries 10 marks)

Q18: Given an array A of positive integers of size N. You are required to reverse every subarray group of size K. Formation of subarray groups will start from the 0th Index of the array, e.g., if the array is of size 5 and K is 3, it will have two subarray groups. The first group will consist of a [0], a[1], and a[2]. The second group will consist of a [3] and a [4].

Note: If at any instance, there are no more subarrays of size greater than or equal to K, then reverse the last subarray (irrespective of its size). You should not return any array and modify the given array in place.

Input Format

The first line contains an integer N denoting the size of the array A and integer K denoting the size of subarray. The next line contains n elements of the array.

Output format

Reverse every subarray group of size K and print the array.

Constraints

$1 \leq N, K \leq 10^4$

$1 \leq A[i] \leq 10^4$

Example

Sample Input

5 3
1 2 3 4 5

Sample Output

3 2 1 5 4

Sample test case explanation

First group consists of elements 1, 2, 3. Second group consists of 4,5.

Test case 1	Test case 2	Test case 3	Test Case 4	Test case 4
Input 10 9 66 87 99 85 50 93 98 84 14 47	Input 50 14 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	Input 8 4 12 34 67 5 90 1 35 78	Input 5 3 15 20 25 30 35	Input 10 5 1 2 3 4 5 6 7 8 9 10
Output 14 84 98 93 50 85 99 87 66 47	Output 20 88 83 44 47 14 84 98 93 50 85 99 87 66 20 38 1 24 64 1 41 5 62 98 13 37 7 71 70 63 72 83 69 1 16 66 3 71 1 57 37 72 20 79 78 3 14 14 29 54	Output 5 67 34 12 78 35 1 90	Output 25 20 15 35 30	Output 5 4 3 2 1 10 9 8 7 6

SOLUTION

```
import java.util.*;
class Main{
          static void reverse(ArrayList<Integer> arr, int n,int left, int right)
          {
            while (left < right) {
                int temp = arr.get(left);
                arr.set(left, arr.get(right));
                arr.set(right, temp);
                left+=1;
                right-=1;
            }
          }
          static void reverseInGroups(ArrayList<Integer> arr, int n, int k) {
        for (int i = 0; i < n; i += k) {
            if(i+k < n){
                reverse(arr,n,i,i+k-1);
            }
            else{
                reverse(arr,n,i,n-1);
            }
        }
          }
  public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int n = scanner.nextInt();
        int k = scanner.nextInt();
        ArrayList<Integer> arr=new ArrayList<>();
        for (int i = 0; i < n; i++) {
            arr.add(scanner.nextInt());
        }
        reverseInGroups(arr, n, k);
        for(int e: arr)
        {
          System.out.print(e + " ");
        }
        scanner.close();
          }
}
```