**General Instructions:**

* **Follow the instructions given in each section.**
* **Make sure that you attempt the questions in order.**

**SECTION-A (10\*1 mark=10 marks)**

***(All questions are compulsory)***

1) Which of the following is not a decision making statement?

a) if

b) if-else

c) switch

**d) do-while**

2) In Java, to create a sub class, keyword '\_\_\_\_\_' is used in the class definition.

**a) Extends**

b) Int

c) Float

d) Create

3) Which of the following defines attributes and methods ?

**a) Class**

b) Object

c) Instance

d) Variable

4) Operator\_\_\_\_\_\_\_\_allocates the memory for an object and returns the address of the object for later use.

a) Int

b) Float

**c) New**

d) Real

5) Java does not support multiple\_\_\_\_\_\_\_\_\_\_

a) Polymorphism

b) Parameters

**c) Inheritance**

d) All of these

6) What is the maximum number of levels possible in a Multilevel Inheritance in Java?

a) 8

b) 16

c) 32

**d) No maximum level**

7) With empty parentheses without arguments, a default\_\_\_\_\_\_is called.

a) Function

**b) Constructor**

c) Class

d) Method

8) At time of array initialization which is necessary to specify?

**a) Row**

b) Column

c) Row and Column

d) None of the above

9) Which of the following is correct?

X: Primitive data types are passed by reference.

Y: Java only supports pass by value. With objects, the object reference itself is passed by value and so both the original reference and parameter copy both refer to the same object .

a) X only

**b) Y only**

c) Both are correct

d) Both are incorrect

10) Which of these methods can be used to search an element in a list?

a) find()

b) sort()

c) get()

**d) binaryserach()**

**SECTION-B (5\*2 mark=10 marks)**

***(All questions are compulsory)***

11) public class ControlStatement {

public static void main(String[] args) {

int a = 5;

int b = 5;

if(a>>2 < ++b){

System.out.println("A less than B");

}else{

System.out.println("A greater than B");

}

}

}

​

**a) A less than B**

b) A greater than B

c) Compile time error

12) What is the correct output of this program ?

int i = 70;

for(; i>60; ){

System.out.print(" "+i);

i--;

}

public class LoopExample {

public static void main(String[] args) {

int i = 70;

for(; i>60; ){

System.out.print(" "+i);

i--;

}

}

}

​

a) Run time Error

b) Compile time error

**c) 70 69 68 67 66 65 64 63 62 61**

d) 70 69 68 67 66 65 64 63 62

13) What will be the output of the following Java code?

class Output

{

public static void main(String args[])

{

Object obj = new Object();

System.out.print(obj.getclass());

}

}

a) Object

b) class Object

**c) class java.lang.Object**

d) Compilation Error

14) How many times 'Hello' is printed?

public class CppBuzz {

public static void main(String[] args){

for(int i = 0; i<5; i=5 )

{

System.out.println("Hello");

}

}

}

a) 5

b) 4

c) 2

**d) 1**

15) Identify the type of msg variable.

public class TestMain {

public static void main(String[] args) {

String msg = "Hello World"; // msg variable name

System.out.println(msg);

}

}

a) Instance Variable

b) Class Variable

**c) Local Variable**

**SECTION-C(Coding Question) (2x5 marks=5 marks)**

Q16) Given a big positive number x represented as string, find value of x % 11 or x mod 11. Output is expected as an integer.

**Input :** x = 1345

**Output:** 3

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | 231456786543567898765 | 65327684325 | 988764 |
| **Output** | 1 | 2 | 7 |

Solution :

**class Solution**

**{**

**static int xmod11(String x)**

**{**

**int n = x.length();**

**int rem=0;**

**for(int i=0; i<n; i++){**

**rem = ( rem\* 10 +(int)(x.charAt(i) - '0'))%11;**

**}**

**return rem;**

**}**

**}**

**class JavaProgram1 {**

**public static void main(String args[])**

**{**

**public static void main(String args[]) throws IOException {**

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**int t = Integer.parseInt(br.readLine());**

**while(t > 0){**

**String x = br.readLine();**

**Solution ob = new Solution();**

**System.out.println(ob.xmod11(x));**

**t--;**

**}**

**}**

**}**

**}**

Q17) Imagine an imaginary array of length N containing balls. Given 2 arrays color and radius of length N each, where color[i] represents the color of the ith ball while radius[i] represents the radius of ith ball. If two consecutive balls have the same color and size, both are removed from the array. wants to know the length of the final imaginary array.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | N = 3  color[] = {2, 2, 5}  radius[] = {3, 3, 4} | N = 4  color[] = {1, 3, 3, 1}  radius[] = {2, 5, 5, 2} | N = 4  color[] = { 2, 1, 1, 5}  radius[] = { 3, 4, 5, 2} |
| **Output** | 1 | 0 | 4 |

Solution :

**import java.io.\*;**

**import java.util.\*;**

**class IntArray**

**{**

**public static int[] input(BufferedReader br, int n) throws IOException**

**{**

**String[] s = br.readLine().trim().split(" ");**

**int[] a = new int[n];**

**for(int i = 0; i < n; i++)**

**a[i] = Integer.parseInt(s[i]);**

**return a;**

**}**

**public static void print(int[] a)**

**{**

**for(int e : a)**

**System.out.print(e + " ");**

**System.out.println();**

**}**

**public static void print(ArrayList<Integer> a)**

**{**

**for(int e : a)**

**System.out.print(e + " ");**

**System.out.println();**

**}**

**}**

**public class MyClass{**

**public static void main(String[] args) throws IOException {**

**int N=4;**

**int[] color = {1, 3, 3, 1};**

**int[] radius = {2, 5, 5, 2};**

**Solution obj = new Solution();**

**int res = obj.finLength(N, color, radius);**

**System.out.println(res);**

**}**

**}**

**class Solution {**

**public static int finLength(int N, int[] color, int[] radius) {**

**Stack<Integer> stack = new Stack<>(); // create a stack to store ball indices**

**for (int i = 0; i < N; i++) { // iterate over all balls**

**if (!stack.isEmpty() && // check if the stack is not empty and**

**color[i] == color[stack.peek()] && // the current ball has the same color as the top ball in the stack and**

**radius[i] == radius[stack.peek()]) { // the current ball has the same radius as the top ball in the stack**

**stack.pop(); // remove the top ball from the stack**

**} else {**

**stack.push(i); // add the current ball to the stack**

**}**

**}**

**return stack.size(); // return the size of the remaining balls in the stack**

**}**

**}**

**SECTION-D (Coding Question)(1x10 mark=10 mark)**

Q18) You are going to book a taxi. There are infinite number of points 1, 2, 3... on the X axis and your current position is cur. There are N Taxis near you, and the position of those taxis is given as an array pos. Where pos[i] denotes the position of the ith taxi. You are also given an array time. Where time[i] denotes the time taken by the ith taxi to cover 1 unit of distance. Your task is to find the minimum time to board a taxi.

**Input :**

N = 3, cur = 4

pos = [1, 5, 6]

time = [2, 3, 1]

**Output :**

2

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | N = 2, cur = 1  pos = [1, 6]  time = [10, 3] | N = 5, cur = 4  pos = [3, 2,1,6,5]  time = [2, 4,1,5,3] | N = 6, cur = 3  pos = [2, 5,6,8,7,4]  time = [2,3,1,5,6,2] |
| **Output** | 0 | 2 | 2 |

Solution :

**import java.io.\*;**

**import java.util.\*;**

**class IntArray**

**{**

**public static int[] input(BufferedReader br, int n) throws IOException**

**{**

**String[] s = br.readLine().trim().split(" ");**

**int[] a = new int[n];**

**for(int i = 0; i < n; i++)**

**a[i] = Integer.parseInt(s[i]);**

**return a;**

**}**

**public static void print(int[] a)**

**{**

**for(int e : a)**

**System.out.print(e + " ");**

**System.out.println();**

**}**

**public static void print(ArrayList<Integer> a)**

**{**

**for(int e : a)**

**System.out.print(e + " ");**

**System.out.println();**

**}**

**}**

**public class MyClass{**

**public static void main(String[] args) throws IOException {**

**int N=6;**

**int cur=3;**

**int[] pos = {2, 5,6,8,7,4};**

**int[] time = {2,3,1,5,6,2};**

**Solution obj = new Solution();**

**int res = obj.minimumTime(N, cur, pos, time);**

**System.out.println(res);**

**}**

**}**

**class Solution {**

**public static int minimumTime(int N, int cur, int[] pos, int[] time) {**

**int k=Integer.MAX\_VALUE;**

**for(int i=0;i<N;i++){**

**int s=(Math.abs(cur-pos[i]))\*time[i];**

**if(s<k){**

**k=s;**

**}**

**}**

**return k;**

**}**

**}**