Let $C = \frac{1}{3}\sum_{i=1}^{3}(x_i - \bar{x})(x_i - \bar{x})^T$
for the data points x_1, x_2, x_3 is calculated.
Find the trace of C.

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Range

Text Areas : PlainText

Possible Answers :

1.3 to 1.5

Question Number : 128 Question Id : 640653351350 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Short Answer Question

Project data points x_1, x_2, x_3 onto a one dimensional space using PCA. Let z_1, z_2, z_3 denotes the projection of x_1, x_2, x_3 respectively. Calculate the summation of all elements of z_2

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

0

Java

Section Id : 64065322138

Section Number :	8
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	16
Number of Questions to be attempted :	16
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065350406
Question Shuffling Allowed :	No

Question Number : 129 Question Id : 640653351354 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "JAVA"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?
CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406531166106. ✔ Yes

6406531166107. ✖ No

Sub-Section Number :	2
Sub-Section Id :	64065350407
Question Shuffling Allowed :	Yes

Question Number : 130 Question Id : 640653351355 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

```
import java.util.*;
class Address implements Cloneable{
    private String city;
    private int pinCode;
    // constructor here to instantiate city, pinCode
    public void setCity(String c){
        city = c;
    }
}
class Temple implements Cloneable{
    private Address adr;
    private String tempName;
    // constructor here to instantiate adr, tempName
    public Address getAddress(){
        return adr;
    }
    public void setTempName(String t){
      tempName = t;
    }
    public Temple clone() throws CloneNotSupportedException{
        Temple t = (Temple) super.clone();
        return t;
    }
    //Overrides the method toString()
    // to return tempName + ":"+ city + ":" + pinCode;
}
public class Main{
    public static void main(String[] args){
        Temple t1 = new Temple(new Address("Madurai", 625001), "Meenakshi temple");
        try{
            Temple t2 = t1.clone();
            t2.setTempName("Golden temple");
            t2.getAddress().setCity("Amritsar");
            System.out.println(t1);
            System.out.println(t2);
        }
        catch(CloneNotSupportedException e){
            System.out.println("Cloning not supported");
        }
    }
}
```

What will the output be?

Options :

6406531166108. ✖
```
Meenakshi temple:Madurai:625001
Golden temple:Amritsar:625001
```

6406531166109. ✔
```
Meenakshi temple:Amritsar:625001
Golden temple:Amritsar:625001
```

6406531166110. ✖
```
Golden temple:Amritsar:625001
Golden temple:Amritsar:625001
```

6406531166111. ✖
```
Meenakshi temple:Madurai:625001
Golden temple:Madurai:625001
```

Question Number : 131 Question Id : 640653351356 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

```
import java.util.*;
public class Test{
    public static void main(String[] args){
        var bill_march = new LinkedHashMap<String, Integer>();
        bill_march.put("suresh", 100);
        bill_march.put("mukesh", 150);
        bill_march.put("ganesh", 80);
        bill_march.put("pranay", 200);
        var bill_april = new LinkedHashMap<String, Integer>();
        bill_april.put("suresh", 200);
        bill_april.put("mukesh", 100);
        bill_april.put("ganesh", 100);
        bill_april.put("pranay", 100);
        var totalBill = new LinkedHashMap<String, Integer>();

        for(Map.Entry<String, Integer> e : bill_march.entrySet())
            totalBill.put(e.getKey(), e.getValue());

        for(Map.Entry<String, Integer> e : bill_april.entrySet())
            totalBill.merge(e.getKey(), e.getValue(), (x, y) -> y + x);

        System.out.println(totalBill);
    }
}
```

Choose the correct option.

Options :

6406531166112. ✖ It generates runtime exception: `NullPointerException`

6406531166113. ✖ It generates the output:
`suresh=100, mukesh=150, ganesh=80, pranay=200`

6406531166114. ✖ It generates the output:
`suresh=200, mukesh=100, ganesh=100, pranay=100`

6406531166115. ✔ It generates the output:
`suresh=300, mukesh=250, ganesh=180, pranay=300`

Question Number : 132 Question Id : 640653351358 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following code.

```
public class ArrayOperations{
  public <T extends Comparable> T min(T[] arr){
    // code for finding the minimum
  }
  public <T extends Number> T sum(T[] arr){
    // code for finding the sum of elements
  }
  public <T> int count(T[] arr){
    // code for finding the count of elements
  }
}
```

What is the class ArrayOperations converted to after type erasure?

Options :

6406531166120. ✖

```
public class ArrayOperations{
  public Object min(Object[] arr){
    // code for finding the minimum
  }
  public Number sum(Number[] arr){
    // code for finding the sum of elements
  }
  public int count(Object[] arr){
   // code for finding the count of elements
  }
}
```

6406531166121. ✖

```
public class ArrayOperations{
  public Object min(Object[] arr){
    // code for finding the minimum
  }
  public Object sum(Object[] arr){
    // code for finding the sum of elements
  }
  public int count(Object[] arr){
    // code for finding the count of elements
  }
}
```

6406531166122. ✔

```
public class ArrayOperations{
  public Comparable min(Comparable[] arr){
    // code for finding the minimum
  }
  public Number sum(Number[] arr){
    // code for finding the sum of elements
  }
  public int count(Object[] arr){
    // code for finding the count of elements
  }
}
```

```
public class ArrayOperations{
  public Comparable min(Comparable[] arr){
    // code for finding the minimum
  }
  public Integer sum(Integer[] arr){
    // code for finding the sum of elements
  }
  public int count(Object[] arr){
    // code for finding the count of elements
  }
}
```

6406531166123.

Question Number : 133 Question Id : 640653351361 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following program.

```
import java.util.stream.*;
import java.util.*;
class Order{
    private String Id;
    private double amount;
    public Order(String id, double amt){
        this.Id = id;
        this.amount = amt;
    }
    public double getAmount(){
        return amount;
    }
}
public class Test{
    public static void main(String[] args){
        var oList = new ArrayList<Order>();     //LINE 1
        oList.add(new Order("A0000", 1000.0));
        oList.add(new Order("A0001", 600.0));
        oList.add(new Order("A0002", 1200.0));
        oList.add(new Order("A0003", 900.0));
        var elements = oList.stream()
                            .filter((x) -> x.getAmount() > 900)
                            .count();           //LINE 2

        System.out.println(elements);          //LINE 3
    }
}
```

Choose the correct option.

Options :

6406531166132. ✖ It generates compiler error at `LINE 1` and `LINE 2`

6406531166133. ✖ It compiles without any error but generates `NullPointerException` at `LINE 1` and `LINE 2`

6406531166134. ✖ It compiles without any error but generates `NullPointerException` at `LINE 2` and `LINE 3`

6406531166135. ✔ It generates the output:
2

6406531166136. ✖

It generates the output:
3

Question Number : 134 Question Id : 640653351363 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

```
import java.util.*;
class Example{
    ArrayList<Integer> list1 = new ArrayList<Integer>();
    ArrayList<Integer> list2 = new ArrayList<Integer>();
    public boolean property(int num) {
        //return true if num is prime otherwise return false
    }
    public void iterateList(ArrayList<Integer> inputlist) {
        Iterator<Integer> it = inputlist.iterator();
        while (it.hasNext()) {
            int element=it.next();
            if(element%2==0) {
                list1.add(element);
            }
            else {
                if(property(element))
                    it.remove();
                else
                    list2.add(element);
            }
        }
        System.out.println(list1+"\n"+list2);
    }
}
public class IteratorTest {
    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<Integer>();
        for (int i = 2; i < 15; i++)
            list.add(i);
        Example obj = new Example();
        obj.iterateList(list);
    }
}
```

What will the output be?

Options :

6406531166141. ✔
```
[2, 4, 6, 8, 10, 12, 14]
[9]
```

6406531166142. ✖
```
[9]
[2, 4, 6, 8, 10, 12, 14]
```

6406531166143. ✖
```
[2, 4, 6, 8, 10, 12, 14]
[3, 5, 7, 9, 11, 13]
```

6406531166144. ✖
```
[3, 5, 7, 9, 11, 13]
[2, 4, 6, 8, 10, 12, 14]
```

Question Number : 135 Question Id : 640653351364 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

```
import java.util.*;
public class MapEx {
    TreeMap<String, String> map=new TreeMap<String, String>();
    public void addToMap(String key,String value){
        map.put(key, _____________);                    //LINE-1
    }
    public static void main(String[] args) {
        MapEx obj=new MapEx();
        obj.addToMap("India", "Sachin");
        obj.addToMap("India", "Sehwag");
        obj.addToMap("Sri Lanka","Hasaranga");
        obj.addToMap("Sri Lanka", "Asalanka");
        for(Map.Entry<String, String> entry:obj.map.entrySet()) {
            System.out.println(entry.getKey()+"-->"+entry.getValue());
        }
    }
}
```

Choose the correct option to fill in the blank in LINE-1 so that the output is:

```
India--> Sachin Sehwag
Sri Lanka--> Hasaranga Asalanka
```

Options :

6406531166145. ✖ `map.getDefault(key, "")+" "+value`

6406531166146. ✔ `map.getOrDefault(key, "")+" "+value`

6406531166147. ✖ `map.putOrDefault(key, "")+" "+value`

6406531166148. ✖ map.getOrDefault(key)+" "+value

Question Number : 136 Question Id : 640653351365 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

```
class InvalidIndexException extends Exception{
    public InvalidIndexException() {
        super("invalid exception");
    }
}
public class Main {
    public static void getSubString(String a, int s, int e)
                        throws StringIndexOutOfBoundsException  {
        try {
            if(s > e)
                throw new InvalidIndexException();
            else
                System.out.println(a.substring(s, e));
        }
        catch(InvalidIndexException ie) {
            StringIndexOutOfBoundsException ne;
            ne = new StringIndexOutOfBoundsException("string index out of bound");
            ne.initCause(ie);
            throw ne;
        }
    }
    public static void main(String[] args) {
        try {
            getSubString("Java program", 5, 0);
        }
        catch(StringIndexOutOfBoundsException se) {
            System.out.println(se.getMessage());
            System.out.println(se.getCause().getMessage());
        }
    }
}
```

What will the output be?

Options :

6406531166149. ✔
```
string index out of bound
invalid exception
```

6406531166150. ✖
```
invalid exception
string index out of bound
```

6406531166151. ✖
```
string index out of bound
null
```

6406531166152. ✖
```
string index out of bound
string index out of bound
```

Question Number : 137 Question Id : 640653351366 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

```
class AssertPositive{
    public static boolean assertPos(int a){
        assert a > 0: "a should not be negative";    //LINE-1
        return true;
    }
}
class AssertNonZero{
    public static boolean assertNonZero(int b){
        assert b != 0: "b should not be zero";     //LINE-2
        return true;
    }
}
public class AssertionTest {
    public static void main(String[] args) {
        int a = -1;
        int b = 0;
        int result = 0;

        if (AssertPositive.assertPos(a) && AssertNonZero.assertNonZero(b))
            result = a - b;     //LINE-3
        System.out.println(result);
    }
}
```

Choose the most appropriate option regarding the code when executed as:
`java -ea:AssertNonZero AssertionTest`

Options :

6406531166153. ✖ LINE-1 gives AssertionError

6406531166154. ✔ LINE-2 gives AssertionError

6406531166155. ✖ LINE-3 gives AssertionError

6406531166156. ✖ This program generates the output:
-1

Question Number : 138 Question Id : 640653351368 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

A.java:

```
package com.pack1;
public class A {
    private void getShow() {                //METHOD-1
        System.out.println("Show with no param");
    }
    public void getShow(int x) {            //METHOD-2
        System.out.println("Show with int param");
    }
    void getShow(String y) {                //METHOD-3
        System.out.println("Show with string param");
    }
    protected void getShow(Object z) {      //METHOD-4
        System.out.println("Show with Object param");
    }
}
```

B.java:

```
package com.pack1;
public class B {
}
```

C.java:

```
package com.pack2;
import com.pack1.A;
public class C extends A{
}
```

Choose the correct option with respect to METHODs 1, 2, 3 and 4 inside class A

Options :

6406531166161. ✖ Class B can access METHODs 1, 2 and 4
Class C can access METHODs 1, 2 and 4

6406531166162. ✔ Class B can access METHODs 2, 3 and 4
Class C can access METHODs 2, 4

6406531166163. ✖

Class B can access METHODs 1, 2, 3 and 4
Class C can access METHODs 1, 2 and 4

6406531166164. ✖ Class B and C both can access all the four methods.

Question Number : 139 Question Id : 640653351369 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Java code given below.

```
class Test{
    public static void method1() throws Exception {
        System.out.println(10/0);
    }
    public static void method2() throws Throwable {
        try {
            method1();
        }
        catch(Exception e) {
            System.out.println("caught in method2()");
            throw e;
        }
    }
}
public class Example {
    public static void main(String[] args) throws Throwable {
        try {
            Test.method2();
        }
        catch(Exception e) {
            System.out.println("caught in main");
        }
    }
}
```

Choose the correct option.

Options :

6406531166165. ✖

This program generates the output:

```
caught in main
caught in method2()
```

6406531166166. ✖

This program generates the output:

```
caught in method2()
```

6406531166167. ✔

This program generates the output:

```
caught in method2()
caught in main
```

6406531166168. ✖ This program terminates abnormally due to unhandled exceptions.

Sub-Section Number : 3

Sub-Section Id : 64065350408

Question Shuffling Allowed : Yes

Question Number : 140 Question Id : 640653351362 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

Consider the Java code given below.

```
import java.util.*;
public class Example {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<String>();
        list.add("Java");
        list.add("Programming language");
        list.add("IITM");
        list.add("Chennai");
        *****CODE BLOCK*****
        for(String str:list) {
            set1.add(str);
            set2.add(str);
        }
        for(String str:set1) {
            System.out.print(str+" ");
        }
        System.out.println();
        for(String str:set2) {
            System.out.print(str+" ");
        }
    }
}
```

Choose the correct option(s) to fill in CODE BLOCK so that the program always generates the following output:

```
Java Programming language IITM Chennai
Chennai IITM Java Programming language
```

Options :

6406531166137. ✖
```
Set<String> set1 = new TreeSet<String>();
Set<String> set2 = new LinkedHashSet<String>();
```

6406531166138. ✔
```
Set<String> set1 = new LinkedHashSet<String>();
Set<String> set2 = new TreeSet<String>();
```

6406531166139. ✖
```
Set<String> set1 = new HashSet<String>();
Set<String> set2 = new TreeSet<String>();
```

6406531166140. ✖
```
Set<String> set1 = new LinkedHashSet<String>();
Set<String> set2 = new HashSet<String>();
```

Sub-Section Number : 4

Sub-Section Id : 64065350409

Question Shuffling Allowed : Yes

Question Number : 141 Question Id : 640653351357 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Select Question

Which of the following statements can count the number of integers between 50 and 100 that are not divisible by 2?

Options :

6406531166116. ✔
```
Stream.iterate(50, n -> n + 1).limit(50).filter(n -> n % 2 != 0)
.count();
```

6406531166117. ✖
```
Stream.iterate(50, n -> n + 1).filter(n -> n % 2 != 0)
.count();
```

6406531166118. ✔
```
Stream.iterate(50, n-> n <100, n -> n + 1).filter(n -> n % 2 != 0)
.count();
```

6406531166119. ✖
```
Stream.filter(50, n-> n <100, n -> n + 1).iterate(n -> n % 2 == 0)
.count();
```

Question Number : 142 Question Id : 640653351359 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Select Question

The code given below should print `true` if the sum of elements of the array is positive, else `false`. Identify the header(s) for function `checkSum` (at `LINE 1`) such that the code prints either true or false, based on value of the `sum`.

```
interface Summable{
    public double sum();
}
class ArrayOperations<T extends Number> implements Summable{
    // instance variables and constructors here
    private T[] arr;
    public ArrayOperations(T[] y){
      this.arr = y;
    }
    public double sum(){ //Finds the sum of elements in arr
      double sum = 0;
      for(T x : arr){
        sum = sum + x.doubleValue();
      }
      return sum;
    }
}

public class Test {
    //LINE 1 {
      if(ob.sum() > 0){
        return true;
      }
      return false;
    }
    public static void main(String[] args){
        Double[] r = {12.3, 14.6, 34.5, 67.0};
        ArrayOperations<Double> d = new ArrayOperations<Double>(r);
        System.out.println(checkSum(d));
    }
}
```

Options :

6406531166124. ✔ `public static <T extends Summable> boolean checkSum(T ob)`

6406531166125. ✔ `public static boolean checkSum(Summable ob)`

6406531166126. ✖ `public static boolean checkSum(T ob)`

6406531166127. ✖ `public static <T> boolean checkSum(T ob)`

Question Number : 143 Question Id : 640653351360 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Select Question

Consider the following program.

```
class ComplexNumber<T extends Number>{
    private T a;
    private T b;
    public ComplexNumber(T a_val, T b_val){
        a = a_val;
        b = b_val;
    }
    public T get_a() {
        return a;
    }
    public T get_b() {
        return b;
    }
    public ComplexNumber<Double> subtract(____LINE 1____) {
        ComplexNumber<Double> c1 = new ComplexNumber<>(0.0, 0.0);
        c1.a = this.a.doubleValue() - c.get_a().doubleValue();
        c1.b = this.b.doubleValue() - c.get_b().doubleValue();
        return c1;
    }
}
public class Test{
    public static void main(String args[]) {
        ComplexNumber<Integer> c1 = new ComplexNumber<Integer>(5,6);
        ComplexNumber<Integer> c2 = new ComplexNumber<Integer>(3,2);
        ComplexNumber<Double> c3 = c1.subtract(c2);
        System.out.println(c3.get_a() + "," + c3.get_b());
    }
}
```

Choose all the options which can be used in place of LINE 1 so that the code outputs the difference between two complex numbers.

Options :

6406531166128. ✔ `ComplexNumber<T> c`

6406531166129. ✖ `ComplexNumber<Number> c`

6406531166130. ✔ `ComplexNumber<?> c`

6406531166131. ✖ `ComplexNumber<Object> c`

Question Number : 144 Question Id : 640653351367 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Select Question

Consider the three Java programs given below.

Student.java:

```
package college;
public class Student {
    private String name, college;
    public Student(String name, String college) {
        this.name = name; this.college = college;
    }
    public String getName() {
        return name;
    }
    public String toString() {
        return name+" "+college;
    }
}
```

Student.java:

```
package dance;
public class Student {
    String name, type;
    public Student(String name, String type) {
        this.name = name; this.type = type;
    }
    public String getType() {
        return type;
    }
    public String toString() {
        return name+" "+type;
    }
}
```

Test.java:

```
package pack3;
public class Test {
    public static void main(String[] args) {
        ******CODE BLOCK******
        System.out.println(obj1);
        System.out.println(obj2);
        System.out.println(obj3);
    }
}
```

Choose the correct option(s) to fill in the CODE BLOCK so that the output is:

```
Sam IITM
Sam Traditional
Ram Traditional
```

Options :

6406531166157. ✔

```
var obj1 = new college.Student("Sam", "IITM");
var obj2 = new dance.Student(obj1.getName(), "Traditional");
var obj3 = new dance.Student("Ram", obj2.getType());
```

6406531166158. ✖

```
var obj1 = new Student("Sam", "IITM");
var obj2 = new Student(obj1.getName(), "Traditional");
var obj3 = new Student("Ram", obj2.getType());
```

6406531166159. ✔

```
college.Student obj1 = new college.Student("Sam", "IITM");
dance.Student obj2 = new dance.Student(obj1.getName(), "Traditional");
dance.Student obj3 = new dance.Student("Ram", obj2.getType());
```

6406531166160. ✖

```
Student obj1 = new college.Student("Sam", "IITM");
Student obj2 = new dance.Student(obj1.getName(), "Traditional");
Student obj3 = new dance.Student("Ram", obj2.getType());
```

AppDev-2

Section Id :	64065322139
Section Number :	9
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	17
Number of Questions to be attempted :	17
Section Marks :	50
Display Number Panel :	Yes
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	64065350410
Question Shuffling Allowed :	No

Question Number : 145 Question Id : 640653351370 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0