Aditya Degree Colleges

Java Online Training Coding Test_5 Key

Date: 27-04-2020

Program – 1:

Sanath is a good student who accomplishes every task given to him. A new principal came to the college and got to know about Sanath. He wants to check Santh's ability. He has given a sentence to Sanath asked him to find out whether there exists equal number of lowercase, uppercase alphabets, digits and symbols in the given sentence. If so he asked Sanath to say "Equality For Everyone" otherwise he asked him to say "No Equality". Now Sanath is really in a dilema. Help Sanath to prove himself worthy for the challenge given by principal.

Input:

A single line of sentence

Output:

A single line either displaying "Equality For Everyone" or "No Equality"

Sample Input 1:

aB1$

Sample Output 1:

Equality For Everyone

Sample Input 2:

ab23$%

Sample Output 2:

No Equality

Source Code:

```
import java.util.*;
class Prog5_1
{	public static void main(String args[])
	{	Scanner sc=new Scanner(System.in);
		int l=0,u=0,d=0,s=0;
		String str=sc.next();
		char x[]=str.toCharArray();
		for(int i=0;i<x.length;i++)
		{
			if(Character.isLowerCase(x[i]))
				l++;
			else if(Character.isUpperCase(x[i]))
				u++;
			else if(Character.isDigit(x[i]))
				d++;
			else
				s++;
		}
		if(l==u && u==d && d==s)
			System.out.println("Equality For Everyone");
		else
			System.out.println("No Equality");
	}
}
```

Testcase1:

Input:

a program to find the given string contains equal number of lowercase alphabets, upper case alphabets, digits and symbol123434023489274@#$%^!@#$%^(*&12345667897654333444

Output:

No Equality

Testcase2:

Input:

abcd23$#,*46

Output:

No Equality

Testcase3:

Input:

12ab32bc$#CB*^BC

Output:

Equality For Everyone

Program – 2:

Problem Description

Race is generally organized by distance but this race will be organized by time. In order to predict the winner we will check every 2 seconds. Let's say total race time is 7 seconds we will check for (7-1) seconds. For 7 sec: We will check who is leading at 2 sec, 4 sec and 6 sec.

Participant who is leading more number of times is winner from prediction perspective. Now our task is to predict a winner in this marathon.

Note:

1)At particular time let say at 4th second, top two (top N, in general) participants are at same distance, then in this case both are leading we will increase count for both (all N).

2)And after calculating at all time slices, if number of times someone is leading, is same for two or more participants, then one who come first in input sequence will be the winner.

Ex: If participant 2 and 3 are both leading with same number, participant 2 will be the winner.

Constraints:

1 <= T <= 100

1 <= N <= 100

Input Format

First line contains a single integer N denoting the number of participants

Second line contains a single integer T denoting the total time in seconds of this Marathon.

Next N lines (for each participant) are as follows :

We have T+1 integers separated by space.

First T integers are as follow:

ith integer denotes the number of steps taken by the participant at the ith second.

T+1st integer denotes the Distance (in meters) of each step.

Output

Index of Marathon winner, where index starts with 1.

Example 1

Input

3

8

2 2 4 3 5 2 6 2 3

3 5 7 4 3 9 3 2 2

1 2 4 2 7 5 3 2 4

Output

2

Explanation

3 (No. of candidate)

8 (Total time of Sprint (In seconds))

2 2 4 3 5 2 6 2 3 (data for 1st candidate. First 8 integers denote number of steps per second and last integer denotes distance covered in each step i.e. 3).

3 5 7 4 3 9 3 2 2 (similarly, 2nd candidate's data).

1 2 4 2 7 5 3 2 4 (similarly, 3rd candidate's data).

At time 2: Here 2nd marathoner is leading

12 (2*3+2*3)

16 (3*2+5*2)

12 (1*4+2*4)

At time 4 :Here also 2nd marathoner is leading

33 (2*3+2*3 +4*3+3*3)

38

36

At time 6 : Here 3rd marathoner is leading

54

62

84

Output:

2

Since, 2nd marathoner is leading more number of times, so 2 is the winner.

Source Code:

```
import java.util.*;
class Practice
{
	public static void main(String args[])
	{
		Scanner sc=new Scanner(System.in);
		int users,sprints;
		int data[][],i,j,winner[],weights[];
		users=sc.nextInt();
		weights=new int[users];
		sprints=sc.nextInt();
		data=new int[users][sprints+1];
		winner=new int[users];
		for(i=0;i<users;i++)
		{
			for(j=0;j<sprints+1;j++)
				data[i][j]=sc.nextInt();
		}
		for(i=2;i<sprints;i=i+2)
		{   int max=0;
		    int k=0;
			int index=0,f=0;
			while(k<users)
			{
			  int sum=0;

			for(j=0;j<i;j++)
			{
			    sum=sum+data[k][j]*data[k][sprints];
			}
			System.out.print(sum+" ");
			weights[k]=sum;
			 if(sum>max)
				max=sum;

			 k++;
			}

			for(int m=0;m<users;m++)
			{
				if(weights[m]==max)
				{
					winner[m]++;

				}
			}
		}
```

```
        int max=0,index=0;
       for(i=0;i<users;i++)
                {
                  //System.out.print(winner[i]+" ");
                  if(winner[i]>max)
                        {
                          max=winner[i];
                          index=i;
                        }
                }
                System.out.println(index+1);
        }
}
```

Testcase1:
Input:
3
8
2 2 4 3 5 2 6 2 3
3 5 7 4 3 9 3 2 2
1 2 4 2 7 5 3 2 4
Output
2

Testcase2:
Input:
4
5
2 3 4 2 4 1
3 6 2 4 1 1
6 2 1 4 3 1
4 5 1 2 2 1
Output:
2

Testcase3:
Input:
2
12
1 2 4 5 6 3 7 8 9 1 2 5 2
4 2 1 5 3 6 2 5 2 1 4 9 2
Output:
1

MCQ_Day5_KEY:

Which of the following are true statements
1) Constructors name should be same as class name.
2) If you don't define a constructor for a class, a default parameter constructor is automatically created by the compiler.
3) The default constructor calls super() and initializes all instance variables to default value like 0, null.
4) If we want to parent class constructor, it must be called in first line of constructor.
A) 1
B) 1,2
C) 1,2,3 and 4
D) 1,2 and 4
ANSWER:C

```
class Test
{
  static int a;
  static
  {
    a=4;
    System.out.print("inside static block");
    System.out.print("a = "+a);
  }
  Test()
   {
     System.out.print("inside constructor");
     a=10;
   }
  public static void func()
  {
      a=a+1;
      System.out.print("a = "+a);
   }
  public static void main(String args[])
  {
     Test obj=new Test();
      obj.func();
   }
}
```

A)Run Time Error
B)inside static blocka = 4inside constructora = 11
C)inside static blocka = 4inside constructora = 5
D)inside static blocka = 10inside constructora = 11
ANSWER:B

Determine the output of the following?

```
package pack1;
public class A
{
        protected void m1()
        {
                System.out.println("Method m1 in A");
        }
}

package pack2;
class B extends A
{
        public void m3()
        {
                System.out.println("Method m3 in B");
        }
    public static void  main(String args[])
        {
                B b=new B();
                b.m1();
        }
}
```

A) Compiler Error
B) Method m1 in A
C) Method m3 in B
D) Runtime Error
ANSWER:A

What is not type of inheritance?
A) Single Inheritance
B) Hierarchical Inheritance
C) Multiple Inheritance
D) Double Inheritance
ANSWER:D

```
class Base
{
  public void foo()
   {
     System.out.println("Base");
   }
 }
class Derived extends Base
{
    private void foo()
    {
      System.out.println("Derived");
    }
 }
class Main
  {
    public static void main(String args[])
     {
       Base b=new Derived();
        b.foo();
     }
  }
```

A) Base
B) Derived
C) run-time error
D) Compile time error

ANSWER:D

Determine the output of the following?

```
package pack1;
public class A
{
        private void m1()
        {
                System.out.println("Method m1 in A");
        }
}
class B
{       public void m3()
        {
                System.out.println("Method m3 in B");
        }
```

```
   public static void  main(String args[])
        {
                A a=new A();
                a.m1();
        }
}
```

A) Compiler Error
B) Method m1 in A
C) Method m3 in B
D) Runtime Error

ANSWER:A

Determine the output of the following?

```
package pack1;
public class A
{
        protected void m1()
        {
                System.out.println("Method m1 in A");
        }
}
class B
{
        public void m3()
        {
                System.out.println("Method m3 in B");
        }
    public static void  main(String args[])
        {
                A a=new A();
                a.m1();
        }
}
```

A) Compiler Error
B) Method m1 in A
C) Method m3 in B
D) Runtime Error

ANSWER:B

A class member declared protected becomes a member of subclass of which type?
A) private member
B) static member
C) public member
D) protected member
ANSWER:A

```
class Test
{
  int i;
 }
class Main
{
  public static void main(String args[])
  {
    Test t;
    System.out.println(t.i);
   }
}
```

A) 0
B) Compile time error
C) Garbage value
D) Runtime error
ANSWER:B

```
class Base
{
  public void show()
  {
    System.out.println("Base::show() called");
   }
}
class Derived extends Base
{
  public void show()
  {
   System.out.println("Derived::show() called");
  }
}
```

```
class Main
{
  public static void main(String args[])
   {
     Base b=new Derived();
      b.show();
    }
}
```

A) Derived::show() called
B) Compiler Error
C) None of the above
D) Base::show() called
ANSWER:A

```
class Test
{
   public void display(int x, double y)
   {
        System.out.println(x+y);
   }
   public double display(int p, double q)
   {
       return (p+q);
    }
   public static void main(String args[])
    {
      Test t=new Test();
      t.display(4,5.0);
      System.out.println(t.display(4,5.0));
   }
}
```

What is the output of the program?
A) 9 9
B) Compile time error
C) 9.0 9.0
D) None
ANSWER:B

Which of the following is true about inheritance in java?
1) Private methods are final
2) Protected members are accessible within a package and inherited classes outside the package.
3) Protected methods are final.
4) We cannot override private methods.
A) 1,2 and 4
B) only 1 and 2
C) 1,2 and 3
D) 2,3 and 4
ANSWER:A

```
class A
{
  int i;
  void display()
   {
     System.out.println(i);
    }
}
class B extends A
{
   int j;
   void display()
   {
     System.out.println(j);
   }
}
class inheritance_Demo
{
   public static void main(String args[])
    { B obj=new B();
      obj.i=1;
      obj.j=2;
      obj.display();
    }
}
```

A) 2
B) 0
C) Compilation Error
D) 1
ANSWER:A

```
package main;
class T
 {
   int t=20;
  }
class Main
{
  public static void main(String args[])
  {
    T t1=new T();
    System.out.println(t1.t);
  }
}
```

A) 0
B) Compiler error
C) 20
D) None

ANSWER:C

```
class demo
{
  int a,b;
  demo()
  {
     a=10;
     b=20;
  }
  public void print()
   {
      System.out.print("a= "+a+" b = "+b);
   }
}
class one
{
  public static void main(String args[])
  {
    demo obj1=new demo();
    demo obj2=obj1;
    obj1.a+=1;
    obj1.b+=1;
    System.out.print("values of obj1: ");
    obj1.print();
```

```
    System.out.print("values of obj2: ");
    obj2.print();
   }
}
```
A) values of obj1: a = 11 b = 21values of obj2: a = 10 b = 20
B) Runtime Error
C) Compiler Error
D) values of obj1: a = 11 b = 21values of obj2: a = 11 b = 21
ANSWER:D

Which of this keyword must be used to inherit a class?
A) super
B) this
C) extends
D) extent
ANSWER:C

```
class Alligator
{
 public static void main(String args[])
  {
    int []x[]={{1,2},{3,4,5},{6,7,8,9}};
    int [][]y=x;
    System.out.println(y[2][1]);
   }
}
```
A) 2
B) Compiler Error
C) 3
D) 7
ANSWER:D

```
class Test
{
   int i;
}
class Main
{
   public static void main(String args[])
   {
      Test t=new Test();
      System.out.println(t.i);
```

```
  }
}
```
A) Garbage value
B) Runtime error
C) 0
D) Compile time error
ANSWER:C

What are the legal modifiers which the constructor can be declared with?
A) public, private, final
B) public, final, static, private
C) public, private, protected
D) public, static, private, protected
ANSWER:C

Which of the following is true?
A) A class can extend more than one class
B) A class can extend only one class but many interfaces
C) An interface can implement many interface
D) A class can extend one class and implement many interfaces
ANSWER:D