

Test will be auto submitted in 150 minutes from the test start time

CODING 5

Q - 1

Report Error

Single File Programming Question Marks : 2 Negative Marks : 0

Create a class **NumberConverter** with the required methods to convert between four major number systems (Decimal, Binary, Octal, and Hexadecimal).

Create a **Main** class and call a suitable method using **NumberConverter** object. Get the source and destination number system as a single character from the user along with the number in the main class. Call a suitable method in **NumberConverter** class to convert.

Note : **D** for Decimal, **B** for Binary, **O** for Octal, and **H** for Hexadecimal.

Input format

The first line of input consists of the Number System Code(From).
The second line of input consists of the Number System Code(To).

The third line of input consists of a number.

Output format

The output prints the result after conversion.

Fill your code here

Java (11)

```
1- import java.util.*;
2- class Num{
3-     public String converter(String n,int s,int d){
4-         return Integer.toString(Integer.parseInt(n,s),d);
5-     }
6- }
7- class Main{
8-     public static void main(String[] args){
9-         Num num=new Num();
10-        Scanner sc=new Scanner(System.in);
11-        char s=sc.nextLine().charAt(0);
12-        char d=sc.nextLine().charAt(0);
13-        String input=sc.nextLine();
14-        if((s=='B') || (s=='b'))
15-            s=2;
16-        if((d=='B') || (d=='b'))
17-            d=2;
18-        if((s=='D') || (s=='d'))
19-            s=10;
20-        if((d=='D') || (d=='d'))
21-            d=10;
22-        if((s=='O') || (s=='o'))
23-            s=8;
24-        if((d=='O') || (d=='o'))
```

☐ Provide Custom Input

Next

Test will be auto submitted in 150 minutes from the test start time

CODING 5

Q - 1

Report Error

Single File Programming Question Marks : 2 Negative Marks : 0

Create a class **NumberConverter** with the required methods to convert between four major number systems (Decimal, Binary, Octal, and Hexadecimal).

Create a **Main** class and call a suitable method using **NumberConverter** object. Get the source and destination number system as a single character from the user along with the number in the main class. Call a suitable method in **NumberConverter** class to convert.

Note : **D** for Decimal, **B** for Binary, **O** for Octal, and **H** for Hexadecimal.

Input format

The first line of input consists of the Number System Code(From).
The second line of input consists of the Number System Code(To).

The third line of input consists of a number.

Output format

The output prints the result after conversion.

Fill your code here

Java (11)

```
10- Scanner sc=new Scanner(System.in);
11- char s=sc.nextLine().charAt(0);
12- char d=sc.nextLine().charAt(0);
13- String input=sc.nextLine();
14- if((s=='B') || (s=='b'))
15-     s=2;
16- if((d=='B') || (d=='b'))
17-     d=2;
18- if((s=='D') || (s=='d'))
19-     s=10;
20- if((d=='D') || (d=='d'))
21-     d=10;
22- if((s=='O') || (s=='o'))
23-     s=8;
24- if((d=='O') || (d=='o'))
25-     d=8;
26- if((s=='H') || (s=='h'))
27-     s=16;
28- if((d=='H') || (d=='h'))
29-     d=16;
30- System.out.print(num.converter(input,s,d));
31- }
32- }
33- }
```

☐ Provide Custom Input

Next

Test will be auto submitted in 150 minutes from the test start time

CODING 5

Q - 2

Report Error

Single File Programming Question Marks : 2 Negative Marks : 0

BO Classes

We can use a BO class for computational purposes.

The Stall owners wanted to calculate the total cost of a particular item type for the given timeline. So add a feature in the application to calculate the total cost for the given timeline.

Create a class **ItemType** with the following attributes,

Attribute	Data Type
name	String
deposit	Double
costPerDay	Double

Add appropriate getter/setter, default, and parameterized constructor.

public ItemType(String name, Double deposit, Double costPerDay).

Get the start date and end date (manipulate as Date object) from the stall owners to calculate rent for the particular ItemType. Write a method **calculateCost** in **ItemTypeBO** class.

Method	Method Description
public Double calculateCost(Date startDate, Date endDate, ItemType typeIns)	returns a Double which corresponds to the total cost.

Create a driver class Main to test the above classes.

Note: Strictly adhere to the Object-Oriented Specifications given

Fill your code here

Java (11)

```
1 // You are using Java
2 import java.util.*;
3 import java.io.*;
4 import java.text.DecimalFormat;
5 import java.text.ParseException;
6 import java.text.SimpleDateFormat;
7 import java.text.DateFormat;
8 class ItemType{
9     public String name;
10    public double deposit;
11    public double costperday;
12    public String getname(){
13        return name;
14    }
15    public void setname(String name){
16        this.name=name;
17    }
18    public void setdeposit(double deposit){
19        this.deposit=deposit;
20    }
21    public double getcostperday(){
22        return costperday;
23    }
24 }
```

☐ Provide Custom Input

Back

Next

Test will be auto submitted in 150 minutes from the test start time

CODING 5

Q - 2

Report Error

Single File Programming Question Marks : 2 Negative Marks : 0

BO Classes

We can use a BO class for computational purposes.

The Stall owners wanted to calculate the total cost of a particular item type for the given timeline. So add a feature in the application to calculate the total cost for the given timeline.

Create a class **ItemType** with the following attributes,

Attribute	Data Type
name	String
deposit	Double
costPerDay	Double

Add appropriate getter/setter, default, and parameterized constructor.

public ItemType(String name, Double deposit, Double costPerDay).

Get the start date and end date (manipulate as Date object) from the stall owners to calculate rent for the particular ItemType. Write a method **calculateCost** in **ItemTypeBO** class.

Method	Method Description
public Double calculateCost(Date startDate, Date endDate, ItemType typeIns)	returns a Double which corresponds to the total cost.

Create a driver class Main to test the above classes.

Note: Strictly adhere to the Object-Oriented Specifications given

Fill your code here

Java (11)

```
19 public void setdeposit(double deposit){
20     this.deposit=deposit;
21 }
22 public double getcostperday(){
23     return costperday;
24 }
25 public void setperday(double costperday)
26 {
27     this.costperday=costperday;
28 }
29 public ItemType(){
30     this.name=null;
31     this.deposit=0;
32     this.costperday=0;
33 }
34 public ItemType(String name,double deposit,double costperday){
35     this.name=name;
36     this.deposit=deposit;
37     this.costperday=costperday;
38     System.out.println(this.name);
39     System.out.println(this.deposit);
40     System.out.println(this.costperday);
41 }
42 }
43 class ItemTypeBO{
```

☐ Provide Custom Input

Back

Next

Test will be auto submitted in 150 minutes from the test start time

CODING 5

Q - 2

Report Error

Single File Programming Question Marks : 2 Negative Marks : 0

BO Classes

We can use a BO class for computational purposes.

The Stall owners wanted to calculate the total cost of a particular item type for the given timeline. So add a feature in the application to calculate the total cost for the given timeline.

Create a class **ItemType** with the following attributes,

Attribute	Data Type
name	String
deposit	Double
costPerDay	Double

Add appropriate getter/setter, default, and parameterized constructor.

public ItemType(String name, Double deposit, Double costPerDay).

Get the start date and end date (manipulate as Date object) from the stall owners to calculate rent for the particular ItemType. Write a method **calculateCost** in **ItemTypeBO** class.

Method	Method Description
public Double calculateCost(Date start, Date end, ItemType typeins)	returns a Double which corresponds to the total cost

Create a driver class Main to test the above classes.

Note: Strictly adhere to the Object-Oriented Specifications given

Fill your code here

Java (11)

```
43 - class ItemTypeBO{
44 -     public double calculatecost(Date start,Date end,ItemType typeins){
45 -         long diff=(start.getTime()-end.getTime())/86400000;
46 -         double result=diff*typeins.costperday;
47 -         return result;
48 -     }
49 - }
50 - class Main{
51 -     public static void main(String args[]) throws ParseException
52 -     {
53 -         ItemType i=new ItemType();
54 -         Scanner sc=new Scanner(System.in);
55 -         DecimalFormat dd=new DecimalFormat("0.0");
56 -         i.name=sc.nextLine();
57 -         i.deposit=Double.parseDouble(sc.nextLine());
58 -         i.costperday=Double.parseDouble(sc.nextLine());
59 -         String date1=sc.nextLine();
60 -         String date2=sc.nextLine();
61 -         ItemType i1=new ItemType(i.name,i.deposit,i.costperday);
62 -         Date start=new SimpleDateFormat("dd/MM/yyyy").parse(date1);
63 -         Date end=new SimpleDateFormat("dd/MM/yyyy").parse(date2);
64 -         ItemTypeBO obj=new ItemTypeBO();
65 -         double result=obj.calculatecost(start,end,i1);
66 -         System.out.println(dd.format(Math.abs(result)));
67 -     }
68 - }
```

☐ Provide Custom Input

Back

Next

Test will be auto submitted in 150 minutes from the test start time

CODING 5

Q - 2

Report Error

Single File Programming Question Marks : 2 Negative Marks : 0

BO Classes

We can use a BO class for computational purposes.

The Stall owners wanted to calculate the total cost of a particular item type for the given timeline. So add a feature in the application to calculate the total cost for the given timeline.

Create a class **ItemType** with the following attributes,

Attribute	Data Type
name	String
deposit	Double
costPerDay	Double

Add appropriate getter/setter, default, and parameterized constructor.

public ItemType(String name, Double deposit, Double costPerDay).

Get the start date and end date (manipulate as Date object) from the stall owners to calculate rent for the particular ItemType. Write a method **calculateCost** in **ItemTypeBO** class.

Method	Method Description
public Double calculateCost(Date start, Date end, ItemType typeins)	returns a Double which corresponds to the total cost

Create a driver class Main to test the above classes.

Note: Strictly adhere to the Object-Oriented Specifications given

Fill your code here

Java (11)

```
47 -         return result;
48 -     }
49 - }
50 - class Main{
51 -     public static void main(String args[]) throws ParseException
52 -     {
53 -         ItemType i=new ItemType();
54 -         Scanner sc=new Scanner(System.in);
55 -         DecimalFormat dd=new DecimalFormat("0.0");
56 -         i.name=sc.nextLine();
57 -         i.deposit=Double.parseDouble(sc.nextLine());
58 -         i.costperday=Double.parseDouble(sc.nextLine());
59 -         String date1=sc.nextLine();
60 -         String date2=sc.nextLine();
61 -         ItemType i1=new ItemType(i.name,i.deposit,i.costperday);
62 -         Date start=new SimpleDateFormat("dd/MM/yyyy").parse(date1);
63 -         Date end=new SimpleDateFormat("dd/MM/yyyy").parse(date2);
64 -         ItemTypeBO obj=new ItemTypeBO();
65 -         double result=obj.calculatecost(start,end,i1);
66 -         System.out.println(dd.format(Math.abs(result)));
67 -     }
68 - }
69 - }
```

☐ Provide Custom Input

Back

Next

Test will be auto submitted in 150 minutes from the test start time

CODING 5

Q - 3

Report Error

Single File Programming Question Marks : 2 Negative Marks : 0

Write a program to check whether the given character is a vowel or consonant.

Create two methods namely main method and alph. Create an object in the main method and access the alph method, that performs the above operation.

Input format

The input consists of a character.

Output format

The output prints whether the character is a vowel or consonant. Display the output as shown in the sample output.

Sample testcases

Input 1	Output 1
j	j :consonant
Input 2	Output 2
e	e :vowel
Input 3	Output 3

Fill your code here

Java (11)

```
1 // You are using Java
2 import java.util.*;
3 class Main{
4     void alph(char ch)
5     {
6         if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
7             System.out.println(ch+" :vowel");
8         else
9         {
10             System.out.println(ch+" :consonant");
11         }
12     }
13     public static void main(String args[])
14     {
15         char ch;
16         Scanner sc=new Scanner(System.in);
17         ch=sc.next().charAt(0);
18         Main obj=new Main();
19         obj.alph(ch);
20     }
21 }
```

☐ Provide Custom Input

Back

Next

Test will be auto submitted in 150 minutes from the test start time

CODING 5

Q - 4

Report Error

Single File Programming Question Marks : 2 Negative Marks : 0

Create a class with two methods one to read the elements of an array and the other to find all pairs of elements in an array whose sum is equal to a specified number.

Input format

The first line of the input consists of the value of n.

The next input is the array of elements separated by space.

The last input is the sum value.

Output format

The output prints the pair whose sum is equal to a specified number else prints that there are no pairs.

Refer to the sample outputs for the formatting specifications.

Sample testcases

Input 1	Output 1
5	3 5
1 2 3 4 5	4 4
8	5 3

Fill your code here

Java (11)

```
1 // You are using Java
2 import java.util.*;
3 import java.io.*;
4 class Main{
5     public static void printarray(int arr[],int n,int sum)
6     {
7         int i,j,s=0;
8         for(i=0;i<n;i++)
9         {
10             for(j=0;j<n;j++)
11             {
12                 if (arr[i]+arr[j]==sum)
13                 {
14                     System.out.print(arr[i]+" "+arr[j]);
15                     System.out.println();
16                     s++;
17                 }
18             }
19         }
20         if(s==0)
21             System.out.println("There is no pairs for the given sum.");
22     }
23     public static void main(String args[])
24     {
```

☐ Provide Custom Input

Back

Next

Test will be auto submitted in 150 minutes from the test start time

CODING 5

Q - 4

Report Error

Single File Programming Question Marks : 2 Negative Marks : 0

Create a class with two methods one to read the elements of an array and the other to find all pairs of elements in an array whose sum is equal to a specified number.

Input format

The first line of the input consists of the value of n.
The next input is the array of elements separated by space.
The last input is the sum value.

Output format

The output prints the pair whose sum is equal to a specified number else prints that there are no pairs.

Refer to the sample outputs for the formatting specifications.

Sample testcases

Input 1	Output 1
5	3 5
1 2 3 4 5	4 4
8	5 3

Fill your code here

Java (11)

```
14      System.out.print(arr[i]+" "+arr[j]);
15      System.out.println();
16      s++;
17      }
18      }
19      }
20      if(s==0)
21          System.out.println("There is no pairs for the given sum.");
22      }
23      public static void main(String args[])
24      {
25          int n,i;
26          Scanner sc =new Scanner(System.in);
27          n=sc.nextInt();
28          int arr[]=new int [n];
29          for(i=0;i<n;i++)
30          {
31              arr[i]=sc.nextInt();
32          }
33          int sum=sc.nextInt();
34          printarray(arr,n,sum);
35      }
36      }
37      }
```

☐ Provide Custom Input

Back

Next

Test will be auto submitted in 150 minutes from the test start time

CODING 5

Q - 5

Report Error

Single File Programming Question Marks : 2 Negative Marks : 0

Create two classes a Box class and a Main class, create an object for the Box class in the Main class, and calculate the volume of the box.

Input format

The input consists of the width, height, and depth separated by space.

Output format

The output displays the volume of the box.
If inputs <= 0 then print "Invalid".

Code constraints

Inputs (double type).

Sample testcases

Input 1	Output 1
7.2 8.0 1.1	63.36

Input 2	Output 2
-1.5 -8.5 -88.0	Invalid

Fill your code here

Java (11)

```
1 // You are using Java
2 import java.util.*;
3 class Box{
4     double width;
5     double height;
6     double depth;
7 }
8
9 class Main{
10     public static void main(String args[]){
11         Box myobj=new Box();
12         double vol;
13         Scanner sc=new Scanner(System.in);
14         myobj.width=sc.nextDouble();
15         myobj.height=sc.nextDouble();
16         myobj.depth=sc.nextDouble();
17         if(myobj.width>0 && myobj.height>0 && myobj.depth>0)
18         {
19             vol=myobj.width*myobj.height*myobj.depth;
20             System.out.printf("%.2f",vol);
21         }
22         else
23         {
24             System.out.println("Invalid");
25         }
26     }
27 }
```

☐ Provide Custom Input

Back

Test will be auto submitted in 150 minutes from the test start time

CODING 5

Q - 5

Report Error

Single File Programming Question Marks : 2 Negative Marks : 0

Create two classes a Box class and a Main class, create an object for the Box class in the Main class, and calculate the volume of the box.

Input format

The input consists of the width, height, and depth separated by space.

Output format

The output displays the volume of the box.
If inputs <= 0 then print "Invalid".

Code constraints

Inputs (double type).

Sample testcases

Input 1	Output 1
7.2 8.0 1.1	63.36
Input 2	Output 2
-1.5 -8.5 -88.0	Invalid

Fill your code here

Java (11)

```
5     double height;  
6     double depth;  
7  
8 }  
9 class Main{  
10     public static void main(String args[]){  
11         Box myobj=new Box();  
12         double vol;  
13         Scanner sc=new Scanner(System.in);  
14         myobj.width=sc.nextDouble();  
15         myobj.height=sc.nextDouble();  
16         myobj.depth=sc.nextDouble();  
17         if(myobj.width>0 && myobj.height>0 && myobj.depth>0)  
18         {  
19             vol=myobj.width*myobj.height*myobj.depth;  
20             System.out.printf("%.2f",vol);  
21         }  
22         else  
23         {  
24             System.out.println("Invalid");  
25         }  
26     }  
27 }  
28 }
```

☐ Provide Custom Input

Clear

Compile & Run

Submit Code

Back