1-> 85.)85.MaxMin Sum

Write a program that accepts 3 integer inputs and finds the sum of maximum and minimum

Business Rules :

1) If any/ or all of the input value is negative then print-1.

2) If any two or all the values in the Input are same then print-2.

Example 1:

Input1: 25

Input2: 2

Input3: 95

Output : 97 (Min 2 + Max 95)

Example2:

Input1:-15

Input2: 49

Input3: 5

Output : -1

Create a class named UserProgramCode that has the following static method

public static int sumMaxMin(int input1, int input2, int input3)

Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of 3 integers.

Output is an integer. Refer sample output and business rules

Sample Input 1:

25

2

95

Sample Output 1:

97

Sample Input 2:

-15

49

5

Sample Output 2:

-1

```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;


namespace ConsoleApplication13
{
   class userprogramcode
   {
      public static int sumMaxMin(int ip1, int ip2, int ip3)
      {
         int ans,a,b;
         int[] t1 = new int[3];
         t1[0] =ip1;
         t1[1] =ip2;
```

```
        t1[2] = ip3;
        for (int i = 0; i < 3;i++)
            if (t1[i] <0)
                return-1;
        for (int i = 0; i < 2; i++)
        {
            for (int j = i + 1; j < 3; j++)
            {
                if (t1[i] == t1[j])
                    return -2;
            }
        }
        a = t1.Max();
        b = t1.Min();
        ans = a + b;
        return ans;
    }
}
class Program
{
    static void Main(string[] args)
    {

        int x,y,z,k;
        x  =Convert.ToInt32(Console.ReadLine());
        y  =Convert.ToInt32(Console.ReadLine());
        z =Convert.ToInt32(Console.ReadLine());
       // k = Convert.ToInt32(Console.ReadLine());
        k =userprogramcode.sumMaxMin(x,y,z);
        Console.WriteLine(k);


    }
```

```
    }
}
```

2-> Find GiftVoucher

In a game two dice is thrown. From the sum of the two dice, the player is going to get the gift voucher from the club. Write a program to find the amount of the gift voucher. Print the amount received as gift.

Sum ofTwo Dices---------- Gift Voucher inRs

2,3,6,11 ----------1000

4,7,10----------3000

5,8,9,12----------5000

In the method,

Only Positive number (1-6) should be given as a input numbers. Else return -1.

Include a class UserProgramCode with a static method findGiftVoucher which accepts two integers. The return type (Integer) should return the gift voucher amount. If the any of the inputs is invalid return -1.

Create a Class Program which would be used to accept a positive Integer, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of two integers.

Output consists of an Integer(the gift voucher amount) or a String “Invalid Input” if any of the inputs is invalid.

Refer sample output for formatting specifications.

Sample Input 1:

1

2

Sample Output 1:

1000

Sample Input 2:

1

-2

Sample Output 2:

Invalid Input

```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;

namespace question36
{
   class Program
   {
      static void Main(string[] args)
      {
         int n, c,m;
         n = Convert.ToInt32(Console.ReadLine());
         m = Convert.ToInt32(Console.ReadLine());
```

```
        c = UserProgramCode.findGiftVoucher(n,m);
        Console.WriteLine(c);

    }
}


class UserProgramCode
{
    public static int findGiftVoucher(int a,int b)
    {
      if(a>0 && b>0 && a<7 && b<7)
      {
        if ((a + b == 2) || (a + b == 3) || (a + b == 6) || (a + b == 11))
            return(1000);
        else if ((a + b == 4) || (a + b == 7) || (a + b == 10))
            return(3000);
        else if ((a + b == 5) || (a + b == 8) || (a + b == 9) || (a + b == 12))
            return(5000);

      }
      else return(-1);
      return 0;
    }

}

}
```

3-> Unique even sum: Write a program to remove all duplicate elements from an input array and return the sum of all even numbers.

Example :
Input: {1,2,7,2,4,8,9,6,8}
After removing duplicates : {1,7,4,9,6}
Output: 4+6 = 10

Exception Rules:

If there is no even number in the input, return -1. if

input contains negative numbers, then return-2.

Include a class UserProgramCode with a static method **addUniqueEven** which accepts an integer array. The return type is an integer as given in the above statement.

Create a Class Program which would be used to accept Input array and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of n+1 values. The first value corresponds to size of the array. The next n numbers contains the integer values.

Output consists of a integer as mentioned in the problem statement. Refer

sample output for formatting specifications.

Sample Input 1:

9

1

2

7

2

4

8

9

6

8

Sample Output 1:

10

Sample Input 2:

5

1

3

5

7

9

Sample Output 2:

-1

Sample Input 3:

4

1

-2

6

8

Sample Output 3:

-2

Ans->
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace l2ndl3
{
    class Usercode
    {
        public static int counter(int[] a)
        {
            int flag = 0;
            List<int> li = new List<int>();
            List<int> li1 = new List<int>(a);
            int i;
            for (i = 0; i < a.Length; i++)
            {
                for (int j = i + 1; j < a.Length; j++)
                {
                    if (a[i] == a[j])
                    {
                        li.Add(a[i]);
                    }
                }
            }

            li1 = li1.Distinct().ToList();
            int[] z = li1.Except(li).ToArray();
            for (i = 0; i < a.Length; i++)
            {
                if (z[i] % 2 == 0)
                {
                    flag = flag + z[i];
                }
            }
            return flag;
        }

    }
}
```

4-> TRAIN TARIFF CALCULATION

Ram has to book his train tickets for travelling from Chennai to Pune through an online portal.
The tariffs vary based on parameters such as the Date of booking(DOB) and the Date of Travel(DOT).
Consider the number of days between DOB and DOT as NOD.

The normal ticket cost from Chennai to Pune when the ticket booking is done minimum one month before is Rs 1000 in Sleeper class(SL). The normal ticket cost for AC class ratings are as follows :

First Class AC (1AC):Rs 2500 Second
Class AC(2AC): Rs 2000 Third Class
AC(3AC) : Rs 1500

A. If NOD is from 21 days upto 30 days ,then the tariff is 10% more than normalticket cost
B. If NOD is from 11 days upto 20 days ,then the tariff is 20% more than normalticket cost
C. If NOD is from 4 days upto 10 days ,then the tariff is 30% more than normalticket cost
D.If NOD is upto 3 days ,then the tariff is 40% more than normal ticketcost.

Write a program to calculate the total cost a person has to pay for their booking given their date of booking , the date of travel and the class of travel as input1, input2 and input3 respectively and print the output in the following format.

Your ticket is confirmed and the booking cost is Rs YYYYYY

where YYYYYY is the calculated booking cost. Print the booking cost as an integer. The date of

booking and the date of travel are given as string in the format yyyy.mm.dd **Business rules:**

1. If the date of travel is less than 3 days from the date of booking, then thetickets cannot be booked and print “Short Notice and hence Tickets cannot be booked”.
2. If the date of booking or the date of travel are not in the date format, then print “Improper Date format in the input”.
3. If the date of travel is more than 90 days from the date of booking, then the tickets cannot be booked and print “Long Notice period and hence Tickets cannot be booked”.
4. If the class of travel is other than SL,1AC,2AC or 3AC, then print “Improper classof Travel”.

Create a class named UserProgramCode that has the following static method public static int calculateTrainTariff(string input1, string input2, string input3) Create a class named Program that accepts the inputs and calls the static method present in the UserProgramCode.

Input and Output Format:

Input consists of 3 strings --- input1 (Date of Booking), input2 (Date of Travel) and input3(class of travel). Refer business rules and sample output for formatting specifications.

Sample Input 1 :

2014.05.15
2014.05.25
SL

Sample Output 1 :

1200

Sample Input 2 :

201405.15

2014.05.25
SL

Sample Output 2 :

Improper Date format in the input

ANS->
```
using System;
usingSystem.Collections.Generic;
usingSystem.Linq;
usingSystem.Text;

namespace train_tariff_calculation
{
    class userprogramcode
    {
        public static int train(string b, string j, string c)
        {
int cost = 0; DateTime dt;
DateTime dt1;
            bool x = DateTime.TryParseExact(b, "yyyy.MM.dd", null,
 System.Globalization.DateTimeStyles.None, out dt);
            bool y = DateTime.TryParseExact(j, "yyyy.MM.dd", null,
 System.Globalization.DateTimeStyles.None, out dt1);
if (!x && !y)
{
                return -1;
```

```
}

int z = (dt - dt1).Days;
Console.WriteLine(z);

if (z <3)
       return -2;
        else if (z > 90)
      return -3;
if (c != "SL" && c != "1AC" && c != "2AC" && c != "3AC")
                        return -4;
if (z > 30 && z <= 90)
{
                        if (c == "SL")
                              cost = 1000; else if
                        (c =="1AC")
                              cost = 2500; else if
                        (c =="2AC")
                              cost = 2000; else if
                        (c =="3AC")
                              cost = 1500;


}
else if (z >=21 && z <= 30)
{
                        if (c == "SL")
                              cost = (int)(1.10*1000); else
                        if (c =="1AC")
                              cost = (int)(1.10*2500); else
                        if (c =="2AC")
                              cost = (int)(1.10*2000); else
                        if (c =="3AC")
                              cost = (int)(1.10*1500);


}
else if (z >= 11 && z <= 20)
{
                        if (c == "SL")
                              cost = (int)(1.20 *1000); else
                        if (c =="1AC")
                              cost = (int)(1.20 *2500); else
                        if (c =="2AC")
                              cost = (int)(1.20 *2000); else
                        if (c =="3AC")
                              cost = (int)(1.20 * 1500);


}
else if (z >= 4 && z <= 10)
{
                        if (c == "SL")
                              cost = (int)(1.30 *1000); else
                        if (c =="1AC")
                              cost = (int)(1.30 *2500); else
                        if (c =="2AC")
                              cost = (int)(1.30 * 2000);
```

```
                    else if (c == "3AC")
                        cost = (int)(1.30 * 1500);


}
else if (z ==3)
{
                    if (c == "SL")
                        cost = (int)(1.40 * 1000); else if
                    (c =="1AC")
                        cost = (int)(1.40 *2500); else if
                    (c =="2AC")
                        cost = (int)(1.40 *2000); else if
                    (c =="3AC")
                        cost = (int)(1.40 * 1500);


}


return cost;


            }
        }
 }
```

5-> **password validation**

Write a method to validate given password. Apply following validations:

1. Minimum length should be 8characters
2. Must contain any one of these three special characters @ or _ or#
3. May contain numbers oralphabets.
4. Should not start with special character ornumber
5. Should not end with specialcharacter

Include a UserProgramCode with a static method validatePassword.The method must return an integer 1 or -1. if it returns 1 then print a message "Valid Password". If the method returns -1 then print a message "Invalid Password".

Create a class Program which gets a string as input and calls the static method validatePassword present in the UserProgramCode.

Input and Output Format:
Input is a string which is the password.
Output is also a string which prints a message "Valid Password" or "Invalid Password".

Sample Input 1:
#bzdfh123c

Sample Output 1:
InvalidPassword

Sample Input 2:
jgu_123dfsd3

Sample Output 2:
Valid Password

```
namespace PASSWORD
{
   class Program
   {
      static void Main(string[] args)
      {
         string s=Console.ReadLine();
         if (Regex.IsMatch(s, @"(^(([a-zA-Z]))(?=.*[\d])(?=.*[a-zA-z])(?=.*[@#_])([a-zA-z0-
9_#@]{8,})([A-Za-z0-9])$)"))
         {
            Console.WriteLine( "valid");

         }
         else
           Console.WriteLine( "invalid");


        }
      }
```

```
    }
```

```
public static int user(string st)
        {

            int count = 0;
            if (st.Length >= 8)
            {
                if (st.Substring(0, 1) == "@" || st.Substring(0, 1) == "_" || st.Substring(0, 1) == "#" ||
st.Substring(st.Length - 1) == "@" || st.Substring(st.Length - 1) == "_" || st.Substring(st.Length - 1) == "#")
                {
                    Console.WriteLine('a');
                    return -1;



                }
                else
                {
                    char[] ch = st.ToCharArray();
                    foreach (var item in ch)
                    {
                        if (item == '@' || item == '#' || item == '_')
                        {
                            count++;
                        }
                    }
                    if (count == 1)
                    {
                        return 1;
                    }
                    else
                    {
                        Console.WriteLine('c');
                        return -1;

                    }
                }
            }
            else
            {
                Console.WriteLine('d');
                return -1;
```

```
    }
}
```

Without main function

```
using System;
usingSystem.Collections.Generic; usingSystem.Linq;
usingSystem.Text;
using System.Text.RegularExpressions;

namespace validate_password
{
class userprogramcode
{
public static int user(string st)
{

Regex r = new Regex(@"^[A-Za-z](?=.[A-Za-z])(?=.[0-
9])(?=.*[#@])([a-zA-Z0- 9@#]{8,})[A-Za-z0-9]$");
//Regex r = new Regex(@"^((?=.[A-Za-z])(?=.[0-
9])(?=.*[@#$])([A-Za-z0- 9@#$]{6,20}))$");
//if (Regex.IsMatch(st, @"^((?=.[\d])(?=.[a-zA-
z])(?=.*[@#$])([a-zA-z0- 9$#@]{6,20}))"))
if (r.IsMatch(st)) return 1;
else
return -1;
//if (Regex.IsMatch(st, @"(^(([a-zA-Z]))(?=.[\d])(?=.[a-zA-
z])(?=.*[@#])([a-zA-z0-9#@]{8,})([A-Za-z0-9])$)"))
//          return1;
//else
//          return-1;
```

}

}

}

6-> Next Year Day

Write a program to read a date String in dd/mm/yyyy format and to calculate the day which falls on the same date next year and print it. Note - return the output in small case.

Example:

Input = 13/07/2012
output = saturday

Include a class UserProgramCode with a static method nextYearDay which accepts a String. The return type (String) should return the day which falls on the same date next year. Return -1 in case the format of the date is incorrect.

Create a Class Program which would be used to accept an Integer, and call the static method present in UserProgramCode.

Input and Output Format:

Input consists of a String, date in dd/mm/yyyy format.
Output consists of a String, the the day which falls on the same date next year.

Refer sample output for formatting specifications.

Sample Input:

13/07/2012

Sample Output:

Saturday

```
public static string user(string st)
      {
         DateTime dt = new DateTime();
         DateTime dt1 = new DateTime();
         DateTime.TryParseExact(st, "dd/MM/yyyy", null, System.Globalization.DateTimeStyles.None,
out dt);
         Console.WriteLine(dt.DayOfWeek);
         dt1=dt.AddYears(1);
          string d=(dt1.DayOfWeek).ToString();
          string d1=d.ToLower();
          return d1
```

}