//codility question

static int Solution**(**int N**)**

**{**

string binary **=** Convert**.**ToString**(**N**,** 2**);**

bool start **=** **false;**

int curLength **=** 0**;**

int maxLength **=** 0**;**

**for** **(**int i **=** 0**;** i **<** binary**.**Length**;** i**++)**

**{**

**if** **(**start **==** **false** **&&** binary**[**i**]** **==** '1'**)**

**{**

start **=** **true;**

**}**

**if** **(**start **==** **true** **&&** binary**[**i**]** **==** '0'**)**

**{**

curLength**++;**

**}**

**if** **(**start **==** **true** **&&** binary**[**i**]** **==** '1'**)**

**{**

**if** **(**curLength **>** maxLength**)**

**{**

maxLength **=** curLength**;**

**}**

curLength **=** 0**;**

**}**

**}**

**return** maxLength**;**

**}**

//1) Create a program that will take a 16 digit number from user

//the card number

//4477 4683 4311 3002

//Reverse the number

//2003 1134 3864 7744

//Even position number multiply by 2

//2+0+0+6+1+2+3+8+3+16+6+8+7+14+4+8

//sum up the 2 digit numbers

//2+0+0+6+1+2+3+8+3+7+6+8+7+5+4+8

//70 % 10 = 0

static void Take16DigitsFromUserAndDoStuff**()**

**{**

Console**.**WriteLine**(**"Please enter your 16 digit card number:"**);**

string cardNum **=** Console**.**ReadLine**().**Replace**(**" "**,** ""**);**

**while** **(**cardNum**.**Length **!=** 16**)**

**{**

Console**.**WriteLine**(**"Invalid number. Please enter your 16 digit card number again:"**);**

cardNum **=** Console**.**ReadLine**().**Replace**(**" "**,** ""**);**

**}**

string cardNumReversed **=** ""**;**

**for** **(**int i **=** cardNum**.**Length **-** 1**;** i **>=** 0**;** i**--)**

**{**

//Console.Write(cardNum[i]);

cardNumReversed **+=** cardNum**[**i**];**

**}**

//Console.WriteLine(cardNumReversed);

int sum **=** 0**;**

**for** **(**int i **=** 0**;** i **<** cardNumReversed**.**Length**;** i**++)**

**{**

**if** **((**i **+** 1**)** **%** 2 **==** 0**)**

**{**

string multEven **=** **(**Convert**.**ToInt32**(**cardNumReversed**[**i**].**ToString**())** **\*** 2**).**ToString**();**

**for** **(**int j **=** 0**;** j **<** multEven**.**Length**;** j**++)**

**{**

//Console.Write(Convert.ToInt32(multEven[j].ToString()) + "+");

sum **+=** Convert**.**ToInt32**(**multEven**[**j**].**ToString**());**

**}**

**}**

**else**

**{**

//Console.Write(Convert.ToInt32(cardNumReversed[i].ToString()) + "+");

sum **+=** Convert**.**ToInt32**(**cardNumReversed**[**i**].**ToString**());**

**}**

**}**

bool check **=** sum **%** 10 **==** 0**;**

**if** **(**check**)**

**{**

Console**.**WriteLine**(**"Valid Card"**);**

**}**

**else**

**{**

Console**.**WriteLine**(**"Invalid Card"**);**

**}**

**}**

//2) Take 11 numbers from user and find that one number which is not repeating

//example

//2,3,4,5,1,10,3,2,5,4,1

//10

static void FindNotRepeating**()**

**{**

Console**.**WriteLine**(**"Please enter 11 numbers seperated by commas (e.g. 11,22,33):"**);**

var input **=** Console**.**ReadLine**().**Split**(**','**);**

**while** **(**input**.**Length **!=** 11**)**

**{**

Console**.**WriteLine**(**"Invalid input. Please enter again:"**);**

input **=** Console**.**ReadLine**().**Split**(**','**);**

**}**

string nonRepeatingValue **=** ""**;**

bool check **=** **false;**

**for** **(**int i **=** 0**;** i **<** input**.**Length**;** i**++)**

**{**

check **=** **false;**

**for** **(**int j **=** 0**;** j **<** input**.**Length**;** j**++)**

**{**

**if** **(**i **!=** j **&&** input**[**i**]** **==** input**[**j**])**

**{**

check **=** **true;**

**}**

**}**

**if** **(**check **==** **false)**

**{**

nonRepeatingValue **=** input**[**i**];**

**break;**

**}**

**}**

**if** **(!**check**)**

**{**

Console**.**WriteLine**(**"The non repeating value is " **+** nonRepeatingValue**);**

**}**

**else**

**{**

Console**.**WriteLine**(**"There are no repeating values"**);**

**}**

**}**

//3) Take number from user until the user inserts a negative number.

//Sort and print all the values

//Find the median and mode(If no repeation then no mode)

static void FindMeanAndMode**()**

**{**

Console**.**WriteLine**(**"Enters numbers (enter a negative number to stop):"**);**

bool check **=** **true;**

string inputs **=** ""**;**

**while** **(**check**)**

**{**

int input**;**

**while** **(!**int**.**TryParse**(**Console**.**ReadLine**(),** **out** input**))**

**{**

Console**.**WriteLine**(**"Invalid input. Try again."**);**

**}**

**if** **(**input **>** 0**)**

**{**

inputs **+=** input **+** ","**;**

**}**

**else**

**{**

check **=** **false;**

**}**

**}**

inputs **=** inputs**.**Remove**(**inputs**.**Length **-** 1**,** 1**);**

var arr **=** inputs**.**Split**(**','**);**

var numArr **=** **new** int**[**arr**.**Length**];**

**for** **(**int i **=** 0**;** i **<** arr**.**Length**;** i**++)**

**{**

numArr**[**i**]** **=** Convert**.**ToInt32**(**arr**[**i**]);**

**}**

Array**.**Sort**(**numArr**);**

var output **=** ""**;**

**for** **(**int i **=** 0**;** i **<** numArr**.**Length**;** i**++)**

**{**

output **+=** numArr**[**i**]** **+** ","**;**

**}**

output **=** output**.**Remove**(**output**.**Length **-** 1**,** 1**);**

Console**.**WriteLine**(**output**);**

double middleIndex **=** numArr**.**Length **/** 2.0**;**

**if** **(**numArr**.**Length **%** 2 **!=** 0**)**

**{**

middleIndex **=** Math**.**Round**(**middleIndex**);**

Console**.**WriteLine**(**"The median is " **+** numArr**[**Convert**.**ToInt32**(**middleIndex**)** **-** 1**]);**

**}**

**else**

**{**

Console**.**WriteLine**(**"The median is " **+** **(**numArr**[**Convert**.**ToInt32**(**middleIndex**)** **-** 1**]** **+** numArr**[**Convert**.**ToInt32**(**middleIndex**)])/**2.0**);**

**}**

**if** **(**CheckIfRepeating**(**numArr**))**

**{**

Console**.**WriteLine**(**"No mode as there is no repeating numbers"**);**

**}**

**else**

**{**

var mode **=** numArr**.**GroupBy**(value** **=>** **value)**

**.**OrderByDescending**(group** **=>** **group.**Count**())**

**.**Select**(group** **=>** **group.**Key**)**

**.**First**();**

Console**.**WriteLine**(**"The mode is " **+** mode**);**

**}**

**}**

static bool CheckIfRepeating**(**int**[]** numArr**)**

**{**

**for** **(**int i **=** 0**;** i **<** numArr**.**Length**;** i**++)**

**{**

**for** **(**int j **=** 0**;** j **<** numArr**.**Length**;** j**++)**

**{**

**if** **(**i **!=** j **&&** numArr**[**i**]** **==** numArr**[**j**])**

**{**

**return** **false;**

**}**

**}**

**}**

**return** **true;**

**}**

//4)https://leetcode.com/explore/featured/card/fun-with-arrays/521/introduction/3237/

static int FindNumbers**(**int**[]** nums**)**

**{**

int count **=** 0**;**

**for** **(**int i **=** 0**;** i **<** nums**.**Length**;** i**++)**

**{**

string numStr **=** nums**[**i**].**ToString**();**

**if** **(**numStr**.**Length **%** 2 **==** 0**)**

**{**

count**++;**

**}**

**}**

**return** count**;**

**}**

//5) https://leetcode.com/explore/featured/card/fun-with-arrays/526/deleting-items-from-an-array/3248/

static int RemoveDuplicates**(**int**[]** nums**)**

**{**

**for** **(**int i **=** 0**;** i **<** nums**.**ToList**().**Distinct**().**Count**();** i**++)**

**{**

nums**[**i**]** **=** nums**.**ToList**().**Distinct**().**ToList**()[**i**];**

**}**

**return** nums**.**ToList**().**Distinct**().**Count**();**

**}**