Codility lesson 1 answer:

using System;

// you can also use other imports, for example:

// using System.Collections.Generic;

// you can write to stdout for debugging purposes, e.g.

// Console.WriteLine("this is a debug message");

class Solution {

public int solution(int N) {

// write your code in C# 6.0 with .NET 4.5 (Mono)

string strBinary = Convert.ToString(N, 2);

int newGap = 0;

int biggestGap = 0;

int gapCount = 0;

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

{

if (strBinary[i] == '0')

{

if (gapCount > 0)

gapCount++;

else gapCount = 1;

}

else

gapCount = 0;

if (gapCount > newGap)

newGap = gapCount;

if (strBinary[i] == '1' && newGap > biggestGap)

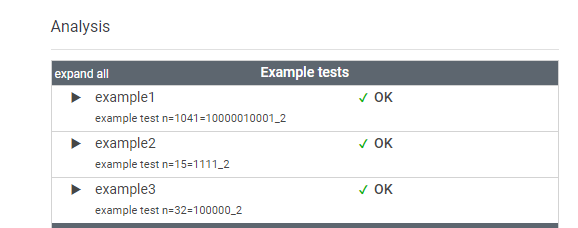
biggestGap = newGap;

}

return biggestGap;

}

}



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

public void validateBankCardQ1()

{

//need to get input and reverse it (add in later)

string cardNum;

Console.Write("Please enter 16 digit number: ");

cardNum = Console.ReadLine();

while (cardNum.Length != 16)

{

Console.WriteLine("\n!!! Please enter number only with exact 16 digit. !!!");

Console.Write("\nPlease enter 16 digit number: ");

cardNum = Console.ReadLine();

}

char[] cardNumArr = cardNum.ToCharArray();

Array.Reverse(cardNumArr);

//this one is hardcode

string reverseInput = new string(cardNumArr);

//Convert to int

int[] inputArr = new int[reverseInput.Length];

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

{

inputArr[i] = (int)(reverseInput[i] - '0');

}

//Even position number multiply by 2

for (int i = 1; i < 16; i = i + 2)

{

int tempValue = inputArr[i];

tempValue = tempValue \* 2;

if (tempValue > 9)

tempValue = (tempValue % 10) + 1;

inputArr[i] = tempValue;

}

//Add up all digits

int total = 0;

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

{

total += inputArr[i];

}

//Validate the card with value

if (total % 10 == 0)

{

Console.WriteLine(total);

Console.WriteLine("This card number is valid.");

}

else

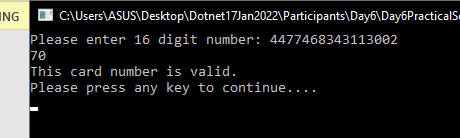
{

Console.WriteLine(total);

Console.WriteLine("This card number is invalid.");

}

}



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

public void repeatingNumbersQ2()

{

int inputNum;

int[] inputNumArr = new int[11];

int matchCount = 0;

Console.WriteLine("Please enter 11 numbers randomly");

for(int i = 0; i < 11; i++)

{

Console.Write("Please enter number no {0} : ", i + 1);

while(!int.TryParse(Console.ReadLine(), out inputNum))

{

Console.WriteLine("Try again. Please enter a number.");

Console.Write("Please enter number no {0} : ", i + 1);

inputNumArr[i] = inputNum;

}

inputNumArr[i] = inputNum;

}

Console.WriteLine("Unique number(s) from 11 numbers: ");

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

{

//bool isDuplicate = false;

matchCount = 0;

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

{

if (inputNumArr[i] == inputNumArr[j])

{

matchCount++;

}

}

if (matchCount == 1)

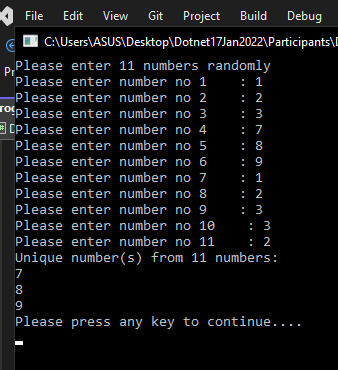
{

Console.WriteLine(inputNumArr[i]);

}

}

}



4) 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)

public void calculatorQ4()

{

List<int> inputNums = new List<int>();

List<int> modeNums = new List<int>();

int inputNum;

int i = 1;

Console.WriteLine("Please enter few numbers for calculation.");

Console.WriteLine("Enter negative value to stop. Eg. -1 ");

Console.WriteLine("=========================================");

do

{

Console.Write("Please enter number no {0} : ", i);

while (!int.TryParse(Console.ReadLine(), out inputNum))

{

Console.WriteLine("Try again. Please enter a number.");

Console.Write("Please enter number no {0} : ", i);

}

//only positive value get saved

if(inputNum >= 0)

inputNums.Add(inputNum);

i++;

} while (inputNum > 0);

//print the values in ascending

inputNums.Sort();

foreach (int num in inputNums)

{

Console.Write(num +", ");

}

////////////////////////////////////////////count median

int[] temp = inputNums.ToArray();

Array.Sort(temp);

int count = temp.Length;

int highestCount = 0;

decimal median = 0;

if (count == 0)

{

Console.WriteLine("Empty collection is not allowed.");

}

else if (count % 2 == 0)

{

// count is even, average two middle elements

int a = temp[count / 2 - 1];

int b = temp[count / 2];

median = (a + b) / 2m;

}

else

{

// count is odd, return the middle element

median = temp[count / 2];

}

///////////////////////////////////////////////////find mode

for (int n = 0; n < temp.Length; n++)

{

//bool isDuplicate = false;

int matchCount = 0;

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

{

if (temp[n] == temp[j])

{

matchCount++;

}

if(matchCount >= highestCount)

{

highestCount = matchCount;

if (!modeNums.Contains(temp[n]))

{

modeNums.Add(temp[n]);

}

}

}

}

//display Median and Mode

Console.WriteLine("\n\nMedian is {0}", median);

Console.Write("Mode = ");

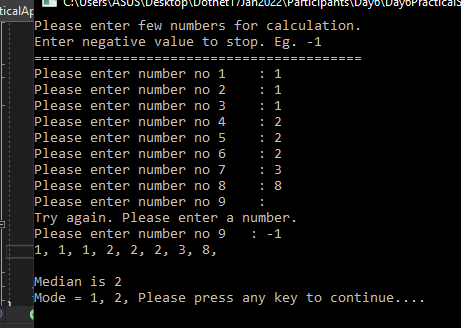
foreach (int num in modeNums)

{

Console.Write(num+ ", ");

}

}



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

public void evenDigitFinderQ5()

{

List<int> inputNums = new List<int>();

int inputNum;

int i = 1;

Console.WriteLine("Please enter few numbers for the program test.");

Console.WriteLine("Enter negative value to stop. Eg. -1 ");

Console.WriteLine("=========================================");

do

{

Console.Write("Please enter number no {0} : ", i);

while (!int.TryParse(Console.ReadLine(), out inputNum))

{

Console.WriteLine("Try again. Please enter a number.");

Console.Write("Please enter number no {0} : ", i);

}

//only positive value get saved

if (inputNum >= 0)

inputNums.Add(inputNum);

i++;

} while (inputNum > 0);

Console.WriteLine("Number(s) with even number of digit: ");

foreach(int num in inputNums)

{

List<int> result = new List<int>();

int tempNum = num;

while (tempNum != 0)

{

result.Insert(0, tempNum % 10);

tempNum = tempNum / 10;

}

int[] numArr = result.ToArray();

if (numArr.Length % 2 == 0)

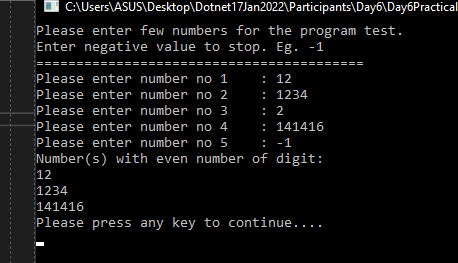
{

Console.WriteLine(num);

}

}

}



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