**Codability - Iteration**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace CodilityApp

{

class Program

{

static void Main(string[] args)

{

Solution sol = new Solution();

int result = sol.solution(1041);

Console.WriteLine(result);

Console.ReadKey();

}

}

class Solution

{

public int solution(int N)

{

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

int longest = 0;

int count = 0;

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

{

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

count++;

else

{

longest = Math.Max(count, longest);

count = 0;

}

}

return longest;

}

}

}

//Output: 5

**Question 1 – 3**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day6SolutionApp

{

class Program

{

static void Main(string[] args)

{

Solution solution = new Solution();

//solution.IsCreditNumber();

//solution.NonRepeatingNumber();

solution.PrintNumberDetails();

Console.ReadKey();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Day6SolutionApp

{

class Solution

{

public static int getDigit(int number)

{

if (number < 9)

return number;

return number / 10 + number % 10;

}

public void IsCreditNumber()

{

long number1;

Console.WriteLine("Please enter the number");

while (!long.TryParse(Console.ReadLine(), out number1) || number1.ToString().Length != 16 )

Console.WriteLine("Please enter a 16 digit number (integer)");

long reverse1 = 0;

int sumEven = 0;

int sumOdd = 0;

int totalSum = 0;

while (number1 > 0)

{

reverse1 = reverse1 \* 10 + number1 % 10;

number1 = number1 / 10;

}

String numEven = reverse1 + "";

for (int i = 15; i >= 0; i -= 2)

sumEven += getDigit(int.Parse(numEven[i] + "") \* 2);

String numOdd = reverse1 + "";

for (int i = 14; i >= 0; i -= 2)

sumOdd += int.Parse(numOdd[i] + "");

totalSum = sumOdd + sumEven;

if ((totalSum) % 10 == 0)

Console.WriteLine("Valid Card");

else

Console.WriteLine("Invalid Card");

}

public void NonRepeatingNumber()

{

int[] numbers = new int[11];

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

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

{

Console.WriteLine("Enter {0} number", i+1);

numbers[i] = Convert.ToInt32(Console.ReadLine());

}

Console.WriteLine("First unique number among the numbers is: ");

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

{

int j;

for (j = 0; j < numbers.Length; j++)

if (i != j && numbers[i] == numbers[j])

break;

if (j == numbers.Length)

Console.WriteLine(numbers[i]);

}

}

public void PrintNumberDetails()

{

int[] numbers = new int[15];

int i = 0;

int input = 0;

int median;

int count = 1;

Console.WriteLine("Please enter upto 15 numbers (stop when negative value is inserted)");

while(i < numbers.Length)

{

Console.WriteLine("Please enter the {0} number", i+1);

input = Convert.ToInt32(Console.ReadLine());

if (input < 0)

break;

else

{

numbers[i] = input;

i++;

count++;

}

}

// Calculate Median

if (count % 2 == 0)

median = (numbers[(count / 2) - 1] + numbers[(count / 2)]) / 2;

else

median = numbers[(count / 2)];

// Calculate Mode...

int[] containers = new int[count+1];

for (int k = 0; k < containers.Length; k++)

containers[k] =numbers[k];

int mode = containers.GroupBy(v => v)

.OrderByDescending(g => g.Count())

.First()

.Key;

Console.WriteLine("------------------------");

Console.WriteLine("The numbers entered are:");

Array.Sort(numbers);

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

{

if (numbers[j] > 0)

Console.WriteLine(numbers[j]);

}

Console.WriteLine("------------------------");

Console.WriteLine("Median: {0}", median);

Console.WriteLine("Mode: {0}", mode);

Console.WriteLine("------------------------");

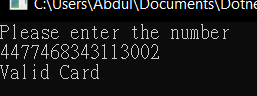
}

}

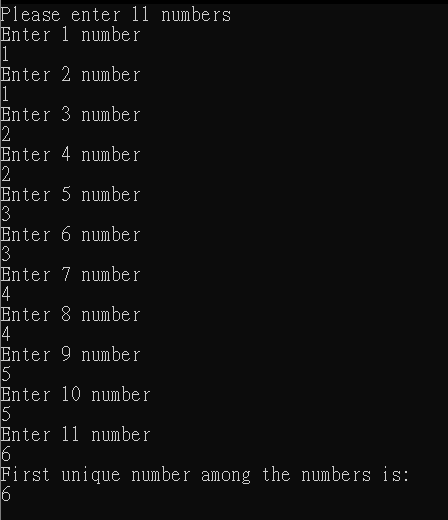
}

**//Output**

**Question 1**



**Question 2**



**Question 3**

