RADI SOFTWARE DO BRASIL LTDA. RDI

Knowledge Evaluation C/C++, C#

Name:
Date: 26/10/2020

- You have **90 minutes** to complete the test.
- Your answers should be given in the language specified in each problem. The code provided in the response must compile and run.
- Each problem provides a code model in the specified language with a generic implementation of input and output, as well as a function that should implement the solution of the problem.
- Write your comments in English.
- For C# problems consider implementing some unit tests;
- Write down your assumptions whenever necessary for better understanding.
- Environment to execute the test (MS Visual Studio 2017) should be previously installed.

Good luck.

Problem #1

Given the time in numerals convert it into words, as shown below:

5:00	five o’ clock
5:01	one minute past five
5:10	ten minutes past five
5:15	quarter past five
5:30	half past five
5:40	twenty minutes to six
5:45	quarter to six
5:47	thirteen minutes to six
5:28	twenty eight minutes past five

At *minutes* = 0, use ***o' clock***. For 1 ≤ *minutes* ≤ 30, use ***past***, and for 30 < *minutes* use ***to***. Note the space between the apostrophe and *clock* in *o' clock*. Write a program which prints the time in words for the input given in the format described.

Function Description

Create a **C++** console solution in Visual Studio using the template below and implement the *timeInWords* function. It should return a time string as described.

timeInWords has the following parameter(s):

- *h*: an integer representing hour of the day
- *m*: an integer representing minutes after the hour

Input Format

The first line contains *h*, the hours portion The second line contains *m*, the minutes portion

Constraints

- $1 \leq h \leq 12$
- $0 \leq m < 60$

Output Format

Print the time in words as described.

Sample Input 1

```
5
47
```

Sample Output 1

```
thirteen minutes to six
```

Sample Input 2

```
3
00
```

Sample Output 2

```
three o' clock
```

Sample Input 3

```
7
15
```

Sample Output 3

```
quarter past seven
```

Code – C++

```
#include "pch.h"
#include <iostream>
#include <fstream>

using namespace std;

// Complete the timeInWords function below.
string timeInWords(int h, int m) {


}

int main()
{
    ofstream fout("OUTPUT1.TXT");

    int h;
    cin >> h;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    int m;
    cin >> m;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    string result = timeInWords(h, m);

    fout << result.c_str() << "\n";

    fout.close();

    return 0;
}

```

Problem #2

The operation called right circular rotation on an array of integers consists of moving the last array element to the first position and shifting all remaining elements right one. Given an array of integers, perform the rotation operation a number of times then determine the value of the element at a given position.
For each array, perform a number of right circular rotations and return the value of the element at a given index.
For example, array a = [3, 4, 5], number of rotations k = 2, and indices to check m = [1, 2].
First we perform the two rotations:

[3, 4, 5] ⇨ [5, 3, 4] ⇨ [4, 5, 3]

Now return the values from the zero-based indices 1 and 2 as indicated in the m array.

$a[1]$ = 5
$a[2]$ = 3

Function Description

Create a **C#** console solution in Visual Studio using the template below and implement the *circularArrayRotation* function. It should return an array of integers representing the values at the specified indices.

circularArrayRotation has the following parameter(s):

- a: an array of integers to rotate
- k: an integer, the rotation count
- *queries*: an array of integers, the indices to report

Input Format

The first line contains 3 space-separated integers, n, k, and q, the number of elements in the integer array, the rotation count and the number of queries.
The second line contains n space-separated integers, where each integer i describes array element $a[i]$ (where $0 \leq i < n$).
Each of the q subsequent lines contains a single integer denoting m, the index of the element to return from a.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq a[i] \leq 10^5$
- $1 \leq k \leq 10^5$
- $1 \leq q \leq 500$
- $0 \leq m < n$

Output Format

For each query, print the value of the element at index m of the rotated array on a new line.

Sample Input

```
3 2 3
1 2 3
0
1
2
```

Sample Output

```
2
3
1
```

Explanation

After the first rotation, the array becomes [3, 1, 2].
After the second (and final) rotation, the array becomes [2, 3, 1].
Let's refer to the array's final state as array b = [2, 3, 1]. For each query, we just have to print the value of b_m on a new line:

1. m = 0, $b[0]$ = 2
2. m = 1, $b[1]$ = 3
3. m = 2, $b[2]$ = 1

Code – C#

```
using System.CodeDom.Compiler;
using System.Collections.Generic;
using System.Collections;
using System.ComponentModel;
using System.Diagnostics.CodeAnalysis;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Reflection;
using System.Runtime.Serialization;
using System.Text.RegularExpressions;
using System.Text;
using System;

class Solution {

    // Complete the circularArrayRotation function below.

    static int[] circularArrayRotation(int[] a, int k, int[] queries) {


    }

    static void Main(string[] args) {
        TextWriter textWriter = new StreamWriter("OUTPUT2.TXT", true);

        string[] nkq = Console.ReadLine().Split(' ');

        int n = Convert.ToInt32(nkq[0]);

        int k = Convert.ToInt32(nkq[1]);

        int q = Convert.ToInt32(nkq[2]);

        int[] a = Array.ConvertAll(Console.ReadLine().Split(' '), aTemp => Convert.ToInt32(aTemp))
        ;

        int[] queries = new int [q];

        for (int i = 0; i < q; i++) {
            int queriesItem = Convert.ToInt32(Console.ReadLine());
            queries[i] = queriesItem;
        }

        int[] result = circularArrayRotation(a, k, queries);

        textWriter.WriteLine(string.Join("\n", result));

        textWriter.Flush();
        textWriter.Close();
    }
}

```

Problem #3

Given an array of integers, find and print the maximum number of integers you can select from the array such that the absolute difference between any two of the chosen integers is less than or equal to *1*. For example, if your array is *a* = [1, 1, 2, 2, 4, 4, 5, 5, 5], you can create two subarrays meeting the criterion: [1, 1, 2, 2] and [4, 4, 5, 5, 5]. The maximum length subarray has *5* elements.

Function Description

Create a **C#** console solution in Visual Studio using the template below and implement the *pickingNumbers* function. It should return an integer that represents the length of the longest array that can be created.

pickingNumbers has the following parameter(s):

- *a*: an array of integers

Input Format

The first line contains a single integer *n*, the size of the array *a*.
The second line contains *n* space-separated integers *a*[*i*].

Constraints

- $2 \leq n \leq 100$
- $0 < a[i] < 100$
- The answer will be ≥ 2

Output Format

A single integer denoting the maximum number of integers you can choose from the array such that the absolute difference between any two of the chosen integers is ≤ 1.

Sample Input 1

```
6
4 6 5 3 3 1
```

Sample Output 1

```
3
```

Explanation 1

We choose the following multiset of integers from the array: {4, 3, 3}. Each pair in the multiset has an absolute difference ≤ 1 (i.e., $|4 - 3| = 1$ and $|3 - 3| = 0$), so we print the number of chosen integers, 3, as our answer.

Sample Input 2

```
6
1 2 2 3 1 2
```

Sample Output 2

```
5
```

Explanation 2

We choose the following multiset of integers from the array: {1, 2, 2, 1, 2}. Each pair in the multiset has an absolute difference ≤ 1 (i.e., $|1 - 2| = 1$, $|1 - 1| = 0$ and $|2 - 2| = 0$), so we print the number of chosen integers, 5, as our answer.

Code – C#

```
using System.CodeDom.Compiler;
using System.Collections.Generic;
using System.Collections;
using System.ComponentModel;
using System.Diagnostics.CodeAnalysis;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Reflection;
using System.Runtime.Serialization;
using System.Text.RegularExpressions;
using System.Text;
using System;

class Result
{

    // Complete the 'pickingNumbers' function below.

    public static int pickingNumbers(List<int> a)
    {


    }

}

class Solution
{
    public static void Main(string[] args)
    {
        TextWriter textWriter = new StreamWriter("OUTPUT3.TXT", true);

        int n = Convert.ToInt32(Console.ReadLine().Trim());

        List<int> a = Console.ReadLine().TrimEnd().Split(' ').ToList().Select(aTemp => Convert.ToInt32(aTemp)).ToList();

        int result = Result.pickingNumbers(a);

        textWriter.WriteLine(result);

        textWriter.Flush();
        textWriter.Close();
    }
}

```