CS162FZ Introduction to Computer Science II

Lab 4

2021-4-18

Note: When dealing with recursion you should first try to work out the solution to the simplest version or versions of the problem – this will be your base case. You also need to determine what the recursive step will be to get from the current problem to a simpler version of the problem.

Task1: Minimum Value

Write two Java methods which return the minimum value in an integer array. You should write:

- An iterative method which returns the minimum value in an integer array. In this method you must use either a for or a while loop. You should call this method q1Iterative and it should have a definition as follows: public static int q1Iterative(int[] a) where a is the input array.
- A recursive method which returns the minimum value in an integer array. You should call this method q1Recursive. The definition of the method should be declared as follows: public static int q1Recursive(int[] a, int start) where a is the input array.

You may assume that there will always be at least one element in the array. In the VPL environment you will be provided with a template for both of these questions. You must use the fillArray method to read in the contents of the array to test your code.

Test cases

Input: n m, where n is the length of the array, followed by m numbers

Sample Input: 6 19 190 1990 9 90 1

Output: Each method will print the value which is the minimum value in the integer array

Sample Output:

1

1

Task2: Maths Function

$$a(n) = \begin{cases} a_1 = 2, & n = 1 \\ a_n = 4a_{n-1} - 3n, & n \geq 2 \end{cases}$$

Given the functional definition below for the mathematical function *a(n),* shown above, you are required to write two Java methods – both methods will be in the same class. You will have seen an example like this in your lecture notes from this week's lectures:

- An iterative method which calculates *a(n)* for any positive value of *n*. In this method you **must** use either a for or a while loop. You should call this method `q2Iterative` and it should have a definition as follows: `public static int q2Iterative(int n)`
- A recursive method which calculates *a(n)* for any positive value of *n*. You should call this method `q2Recursive`. The definition of the method should be declared as follows: `public static int q2Recursive(int n)`

Both Java methods should only return the single value of *a(n).* They should not print any other values. You should use the **`Scanner`** class to obtain the single value for *n*. **You must** show the demonstrator for your row a diagram of the recursive calls for **`q2Recursive`** method as part of your solution. This was explained in the lectures and it is important for you to draw this diagram to illustrate your understanding of how recursion works.

Test cases

Input: n, an integer value.

Sample Input: 1

Output: The method will print the value of a(n) for each method (which is a(1) in this case)

Sample Output:

```
2
2
```

Input: n, an integer value.

Sample Input: 5

Output: The method will print the value of a(n) for each method (which is a(5) in this case)

Sample Output:

```
-79
-79
```

Task3: String Reverse

Write a recursive method (no loops) which takes a string and reverses it. You should call this method `reverseRecursive`. The method should take a string and return the string reversed.

Test cases

Input: n, where n is a String

Sample Input: Hello

Output: The String reversed

Sample Output: olleH

Task 4: Sum Digits

Write a recursive method (no loops) which takes in a non-negative integer and returns the sum of all the digits in a number. You should call this method `recursiveSum`.

For Example

recursiveSum(126) would return 9 as 1 + 2 + 6 = 9

Test cases

Input: n, where n is a non-negative integer

Sample Input: 126

Output: The sum of all single digits in a number

Sample Output: Res: 9

Task 5: Array value times 10

Given an integer array of size n write a recursive method (no loops) that checks if the array contains somewhere a value followed in the array by that value times 10. You should call this method `checkArray`. The method should take an array and a starting index and should return a Boolean value. The array should be created using the fillArray() method provided.

For Example

checkArray([1, 2, 20], 0) → true

checkArray([3, 30], 0) → true

checkArray([3], 0) → false

checkArray([1, 10, 40, 6, 60, 60, 54] , 4) → false

Test cases

Input:

n m, where n is the length of the array, followed by m numbers

Sample Input:

3 1 2 20

Output:

A boolean value indicating if the value is followed by the value time 10

Sample Output:

true

Task 6(Due to 2021-04-20, please submit separately)

Using the code template given in Programming4.java and the 4 pieces of code in the folder “Student Code” your task is to use your knowledge of regular expressions to write a piece of code that will correct 4 pieces of Student code. The students were asked to write java code to print all the even numbers between 0 and 50 (inclusive). 4 Students have submitted code and the results are stored in the folder “Student code”. You can award the following marks for the code having each of the following: (possible answers given, they can have other variations)

- Class declaration (10 Marks)
    - .*public class.*
- A main method (10 Marks)
    - .*void main\(String \[\] args\).*
- A syntactically correct for loop (10 Marks)
    - .*for\(.*;.*;.*\).*
- The modulo operator (10 Marks)
    - .*%.*
- A valid print statement (10 Marks)
    - .*System\.out\.print.*

When looking for the above patterns don’t forget that code is indented!

You can pass the filename of each code into the method *codeReader* as a String. That will read the code from one submission. You must do this for all 4 students code. The output should have the student and the mark received. See potential output below. Actual marks are given here for demonstrators to use when marking!

Example Output:

```
Student 1: 50 Marks
Student 2: 30 Marks
Student 3: 30 Marks
Student 4: 10 Marks
```

important Notes:

In Programming4.java you are provided a method that will take in a file name and return each line of the file as a String and pass that into a String array. It is important that you do not change any of the code indicated.

IMPORTANT: The 4 java files must be placed in the same folder as the Programming4.java for the given code to work correctly.