Lab 12: Nested Structures
CSE-020-01, Fall 2019 **Points: 20**

Overview

This week we are going to put together what we have learned in this course and get more practice using nested structures. Every construct we have learned: conditionals, loops and arrays can be used inside one another to make complicated structures.

Loop Related Statements

Break Statement:

Makes the loop end completely

Continue Statement:

Skips the rest of the current iteration of the loop
Goes to the next iteration

You can review Section 4.9 on how to use these statements. They should come in handy when implementing parts of this lab.

Fibonacci

Fibonacci sequence consists of numbers that are the sums of the previous two numbers. Starting from the position 0, we get the sequence as follows:

Position	0	1	2	3	4	5	6	7	8
Number	0	1	1	2	3	5	8	13	21

So, for position 6, we take the value at positions 5 and 4 and sum them up. This will result in the value of 8 (3 + 5).

Getting Started

After starting Eclipse, create a new project called **Lab20_12**. Import `ChooseFunc.java` and `MinMaxReverse.java` from the Lab 12 assignment page into the project.

Part 1: Fill-in – ChooseFunc.java

This program behaves as follows: should ask the user to choose from 4 functions it is able to perform, as behaves as follows:

- It asks the user to choose from 4 functions, and the value of `max`, which is the number of values we wish to compute according to the function choice.
- If the function number is not between 1 – 4 then it prints out an error *once*.
- If the user want to re-enter the inputs (function choice and `max`), it repeats the input operations (and calculations).
- If the user chooses not to repeat the inputs, it asks the user if they wish to see the array's contents, and if so, prints them out as one index and value per line as shown below.
- The program then asks the user if they wish to quit and ends if they so wish.

Your program must produce an output as follows (inputs are shown in **green**):

```
This program supports 4 functions of a max number:
      1. SumAll
      2. SumSquare
      3. Factorial
      4. Fibonacci
Please choose the function you want: 0
```

```
Please enter the max number (> 0): 5
Error: 0 is an invalid choice for function.

Do you wish to repeat your inputs? (1 for yes): 1
Please choose the function you want: 1
Please enter the max number (> 0): 5

Do you wish to repeat your inputs? (1 for yes): 0

Do you wish to print the results? (1 for yes): 1
Results in the array:
      Index  Value
      0      0
      1      1
      2      3
      3      6
      4      10
      5      15

Do you wish to quit the program? (1 for yes): 0

Please choose the function you want: 5
Please enter the max number (> 0): 5
Error: 5 is an invalid choice for function.

Do you wish to repeat your inputs? (1 for yes): 0

Do you wish to print the results? (1 for yes): 1
Results in the array:
      Index  Value
      0      0
      1      0
      2      0
      3      0
      4      0
      5      0

Do you wish to quit the program? (1 for yes): 0

Please choose the function you want: 3
Please enter the max number (> 0): 8

Do you wish to repeat your inputs? (1 for yes): 0

Do you wish to print the results? (1 for yes): 0

Do you wish to quit the program? (1 for yes): 1
```

Part 2: Fill-in – MinMaxReverse.java

This program finds the minimum number and its index, the maximum number and its index in the given array (`arr`). It then places the contents of `arr` in reverse order into another array `rev_arr`. For example, the first element of `arr` will be placed into the last entry of `rev_arr`. Code is already provided in the program to print out the elements of both arrays in opposite order. The values printed in each line should match once you have inserted the proper code.

Your program must produce an output as follows:

```
Min number is -3000 found at index 2
Max number is 50000 found at index 7

Printing out contents of both arrays in opposite order...
```

```
arr[0]=1     is the same as     rev_arr[12]=1
arr[1]=2     is the same as     rev_arr[11]=2
arr[2]=-3000 is the same as     rev_arr[10]=-3000
arr[3]=4     is the same as     rev_arr[9]=4
arr[4]=5     is the same as     rev_arr[8]=5
arr[5]=-100  is the same as     rev_arr[7]=-100
arr[6]=3     is the same as     rev_arr[6]=3
arr[7]=50000 is the same as     rev_arr[5]=50000
arr[8]=700   is the same as     rev_arr[4]=700
arr[9]=2     is the same as     rev_arr[3]=2
arr[10]=4    is the same as     rev_arr[2]=4
arr[11]=6    is the same as     rev_arr[1]=6
arr[12]=8000 is the same as     rev_arr[0]=8000
```

Part 3: (Assessment) Logic Check

Create a Word document or text file named `Part3` that contains answers to the following:

1. Answer the following about `resArr` in `ChooseFunc.java`:
   a. Why is the datatype `long` (instead of `int`)?
   b. Why is it declared with `[max+1]` entries?
2. Provide an alternative expression to `while(true)` in `ChooseFunc.java` wto implement the same behavior where it keeps looping until the user chooses to end.
3. Answer the following about `min` and `max` variables in `MinMaxReverse.java`:
   a. Why is `min` initialized to the biggest positive `int` (`Integer.MAX_VALUE`)?
   b. Why is `max` initialized to the smallest negative `int` (`Integer.MIN_VALUE`)?

What to hand in

When you are done with this lab assignment, submit all your work through CatCourses.

Before you submit, make sure you have done the following:

- Attached the file named `Part3` containing answers to the assessment questions.
- Attached the filled-in `ChooseFunc.java` and `MinMaxReverse.java` files.
- Filled in your collaborator's name (if any) in the "Comments…" text-box at the submission page.

Also, remember to demonstrate your code to the TA or instructor before the end of the grace period.