

Lab 6 – Recursion

Perform three exercises to get hands on experience with recursion

Task 1. Padovan Sequence:

[LP] Part 1: In a file called **Padovan.txt**, write the pseudocode to recursively compute the n^{th} Padovan number¹. A padovan number is an extension to the Fibonacci series that is defined by the relation:

$$P(n) = P(n - 2) + P(n - 3). P(0) = P(1) = P(2) = 1.$$

Clearly state your base case(s).

[LP] Part 2: Implement the pseudocode in a function called **unsigned int padovan(const unsigned int &n)** in a file called **padovan.cpp**.

[LP] Part 3: Debug through your code² to determine in what order the function **padovan** is called and with what parameters. You may choose to use cout statements or note down the steps during the debug process manually. In the file called **Padovan.txt**, list the function calls and returns in the order in which they were made when n is 7. E.g., when n=5, the list of function calls and returns is:

1. In padovan with n = 5
2. In padovan with n = 3
3. In padovan with n = 1
4. Returning 1 with n = 1
5. In padovan with n = 0
6. Returning 1 with n = 0
7. Returning 2 with n = 3
8. In padovan with n = 2
9. Returning 1 with n = 2
10. Returning 3 with n = 5

[HP] Part 4: In the file called **Padovan.txt**, discuss briefly if the recursive implementation is suitable for this function as compared to an iterative implementation.

¹ <https://mathworld.wolfram.com/PadovanSequence.html>

² This tutorial may help if you use the command line: <https://web.eecs.umich.edu/~sugih/pointers/summary.html>

Task 2. Check if a number is a palindrome using recursion:

[LP] Part 1: In a file called `palindrome.txt`, write the pseudocode to recursively check if a number is a palindrome or not. A palindrome is a number that reads the same backwards and forwards. E.g., 1234321. In your implementation, you may not use a separate function to reverse the number.

[LP] Part 2: Once you have the pseudocode, write a function `bool check_palindrome(const int &n)` in a file called `palindrome.cpp` that implements the recursive algorithm you have written. Use the main function to verify the functionality of your function.

[HP] Part 3: Determine the computational complexity of your code and report the same in `palindrome.txt`.

Task 3. Algorithm Analysis, Thoughts and Documentation:

[HP] Extend your implementation of the `arrayList` template and the `linkedList` template from Labs 3-5 to include two functions called `elemType getMinRecursive()` that finds the smallest element in a list using recursion.

Specifications

All tasks have components labeled [LP] and [HP]. If you complete ALL the LP components satisfactorily, you will receive a grade of “low pass” on the lab. If you complete ALL the LP components and 2 of the 3 HP components satisfactorily, you will receive a grade of “high pass”. If you do not meet the criteria for a “low pass”, the submission will be marked as “revision needed”.

What to submit:

Your final submission will need to have the files as follows:

- `Padovan.txt`
- `padovan.cpp`
- `palindrome.txt`
- `palindrome.cpp`
- `arrayList.h`
- `linkedList.h`
- `main.cpp`
- Notes to include all your references.

NOTE: You can look for help on the Internet but refrain from referencing too much. Please cite all your sources in your Notes file.

When to submit:

Submit your lab before **Thursday, March 28th, 11:59pm**. You are strongly advised to submit before Friday, March 22nd, 11:59pm.

When you submit your assignment, you automatically agree to the following statement. If you do not agree, it is your responsibility to provide the reason.

"I affirm that I have neither given nor received unauthorized help in completing this homework. I am not aware of others receiving such help. I have cited all the sources in the solution file."