Behjat Bahmani

Professor Suporn Chenhansa

CS-124-01

2 April 2021

Lab 3

Purpose:

The purpose of this assignment is to know more about stack (last-in, first-out) and queue(first-in, first-out) data structures and their functions, how they work and how to use them to find palindromes. Moreover, I got more familiar with linked list and its use cases.

I used a linked list to store my data in it. Moreover I learnt more about data allocation and deallocation. string manipulation and cleaning raw data. After that I learnt about reading data from console and file and writing in file. In this lab I read more about object oriented programming and its concepts like encapsulation and I used it in the program. At the end I learnt about valgrind and memory-leak and techniques to solve this problem. I used the "RAII" (Resource acquisition is initialization) concept to solve memory-leak in my program.

Plan:

Since I've already implemented stack, queue, linkedlist in an assignment in class with my group, I tried to make them better for this lab. I use stack and queue to find palindromes and linkedlist to store data. Firstly, I greet and introduce my program. Then I ask the user if he wants to try my program. Then user can choose how he wants the program to read input, from file or from console. if user chooses a file, "fromFile" function is called. this function asks the user for the full path of the file. Reads the file line by line and cleans raw input. For each parsed input isPalindrome function is called which tells us if input is palindrome or not. If an input is palindrome it’s pushed in alphabetized order in a linked list we’ve already initialized. Otherwise we pushed it in order in another linked list for non-palindromes. If user chooses to enter input from the console, the fromConsole function is called. The program asks user to enter inputs one by one and also gives user a way to exit the program. This function works like a fromFile function, except it takes input from the console. They both use the same parse and isPalindrome function. If user doesn't choose one of these options, the program exits with a message. At the end of fromFile and fromConsole functions user is asked if he wants to save its results. If yes, the “save” function is called with palindromes and non-palindromes linkedlist that we used to store data in them as arguments.

User is asked to type a filename for the output file. Save function reads the linked lists we passed as arguments when we called the function and writes their contents in the file we specified.

Input:

1. User chooses if he/she wants to try or not (Yes or No).
2. User chooses if he wants to enter a file as input or writes inputs line by line in the console. (f or file for file and p or phrase or phrases for phrase)
   a. If user chooses to enter phrases line by line, he can enter phrases line by line or he can enter “exit” or “quit” to leave the program.
3. User chooses if he wants to store output in a file. (Yes or No)

Functions:

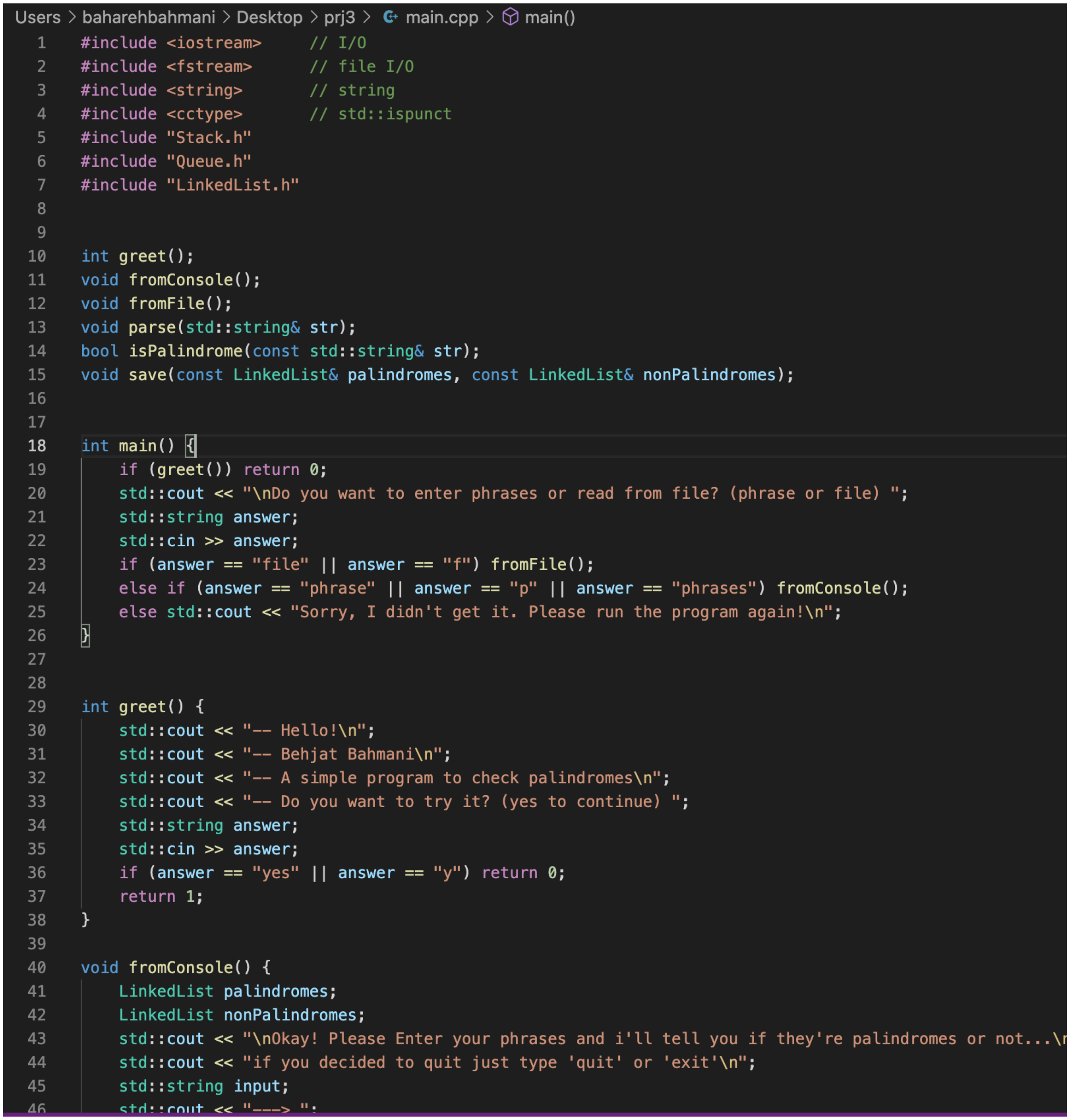
1. Int main(): This is the main function and starting point of our program.
2. Void greet(): greeting and introducing program.
3. Void fromFile(): This function is called if user decided to enter input data from file.
4. Void fromConsole(): This function is called if user decided to write phrases himself in console
5. Void parse(string& str): this function is called every time an input is read to clean and parse raw input strings
6. bool isPalindrome(const string& str): gets cleaned string as argument and check if string is palindrome or not using stack and queue.
    - Member functions:

- Stack:
- a) Void push(char value);
- b) bool empty() const;
- c) Const char& top() const;
- d) Char pop;
- e) Void print () const;
- Queue:
- a) void push(char value);
- b) char pop();
- c) const char& front();
- d) const char& back();
- e) void print() const;

7. void save(const LinkedList& palindromes, const LinkedList& nonPalindromes): it gets stored string the linked list and writes them into the file.

Development process:

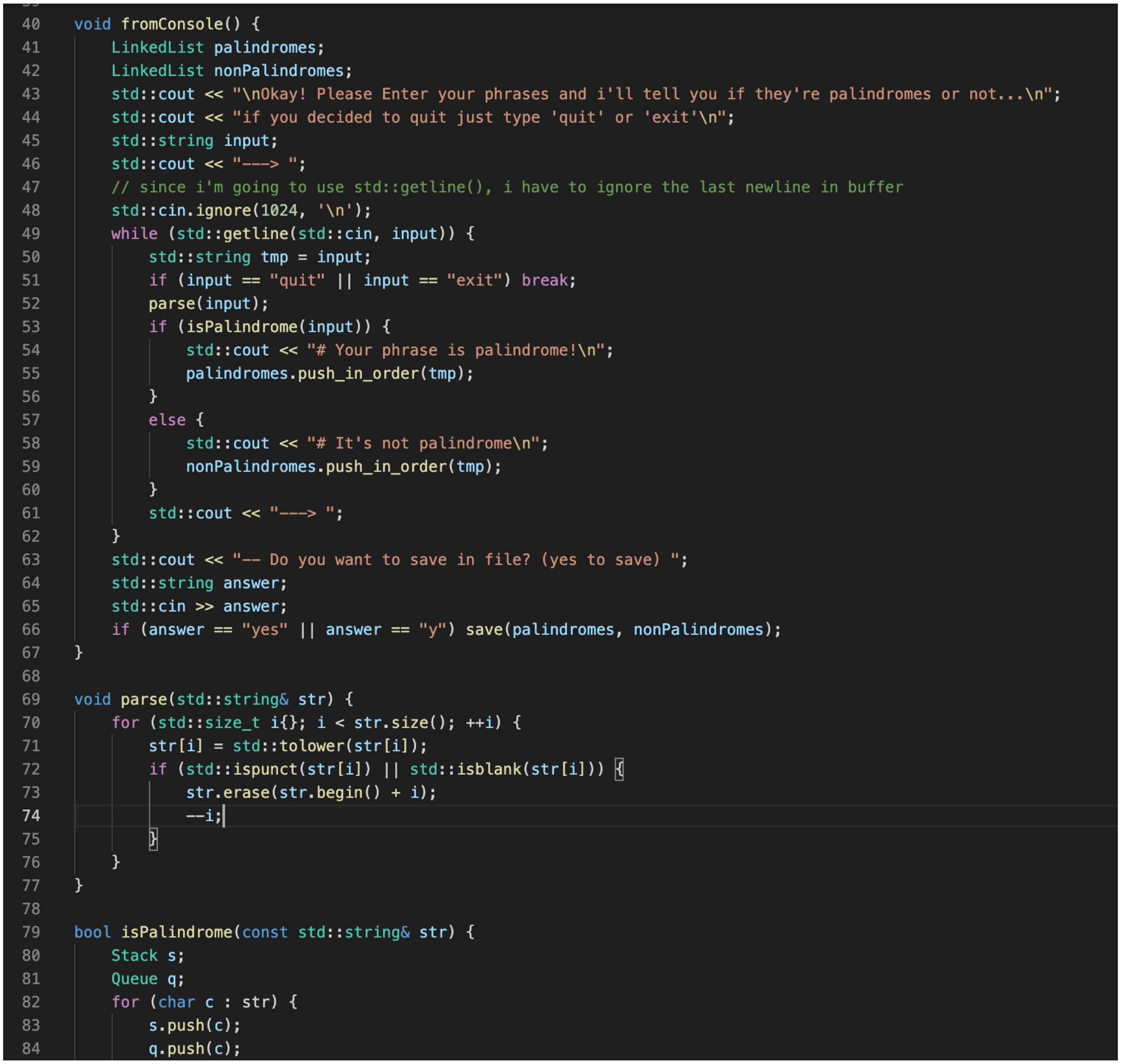
This part I wrote the functions to greet after that ask if the user wants to choose to write a or some phrases or a file. If the user choose phrases and then function fromFile is run and if the user chooses a file and then the function fromConsole runs.



Users > baharehbahmani > Desktop > prj3 > main.cpp > main()

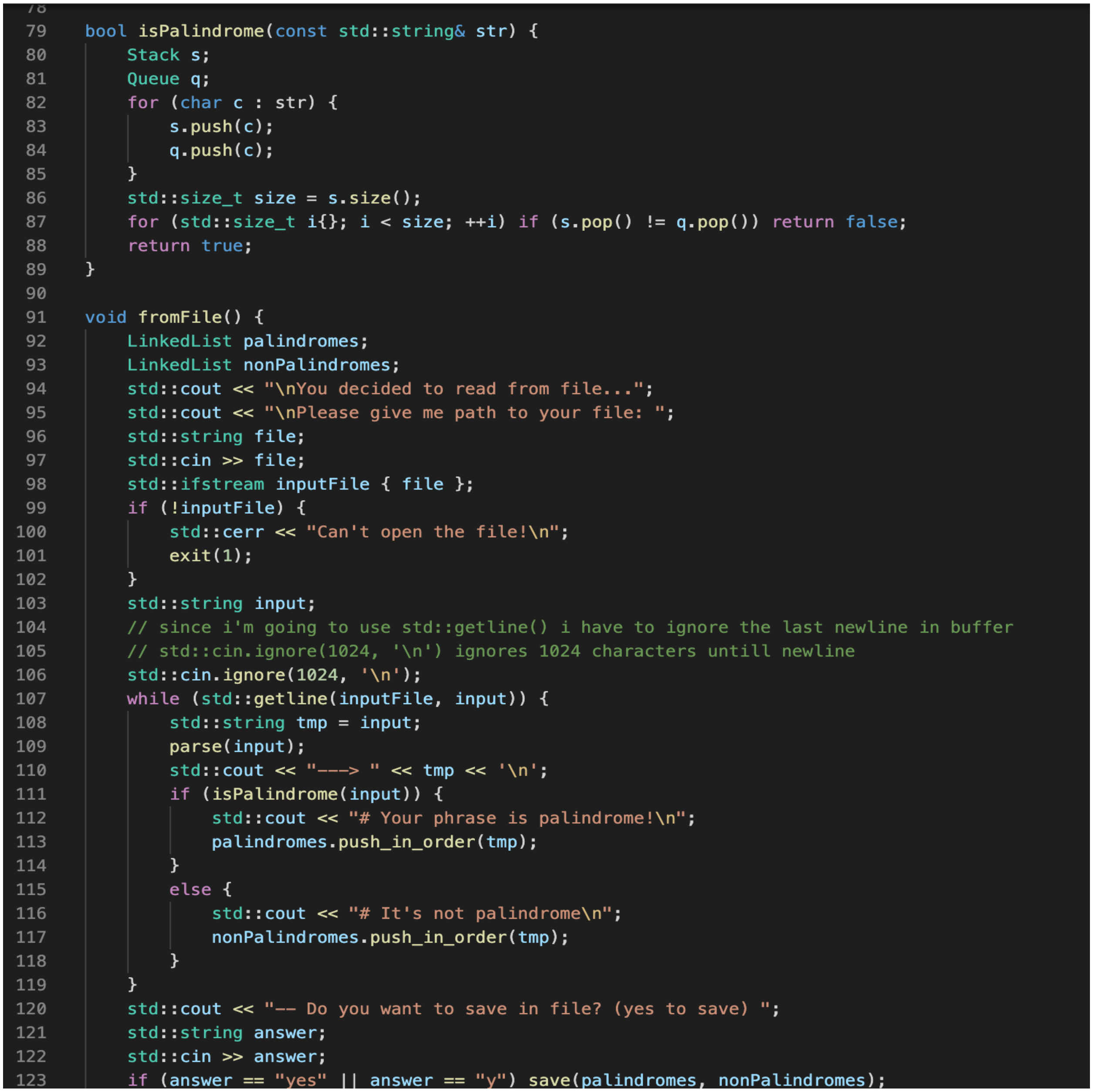
```cpp
#include <iostream>     // I/O
#include <fstream>      // file I/O
#include <string>       // string
#include <cctype>       // std::ispunct
#include "Stack.h"
#include "Queue.h"
#include "LinkedList.h"


int greet();
void fromConsole();
void fromFile();
void parse(std::string& str);
bool isPalindrome(const std::string& str);
void save(const LinkedList& palindromes, const LinkedList& nonPalindromes);


int main() {
    if (greet()) return 0;
    std::cout << "\nDo you want to enter phrases or read from file? (phrase or file) ";
    std::string answer;
    std::cin >> answer;
    if (answer == "file" || answer == "f") fromFile();
    else if (answer == "phrase" || answer == "p" || answer == "phrases") fromConsole();
    else std::cout << "Sorry, I didn't get it. Please run the program again!\n";
}


int greet() {
    std::cout << "-- Hello!\n";
    std::cout << "-- Behjat Bahmani\n";
    std::cout << "-- A simple program to check palindromes\n";
    std::cout << "-- Do you want to try it? (yes to continue) ";
    std::string answer;
    std::cin >> answer;
    if (answer == "yes" || answer == "y") return 0;
    return 1;
}

void fromConsole() {
    LinkedList palindromes;
    LinkedList nonPalindromes;
    std::cout << "\nOkay! Please Enter your phrases and i'll tell you if they're palindromes or not...\
    std::cout << "if you decided to quit just type 'quit' or 'exit'\n";
    std::string input;
    std::cout << "---> ";
```

In this part inside the function fromConsole, I parse the phrase/phrases and runs the function "fromConsole" if the file or phrase isPalindrome or nonPalindrome and then push_in_order. At the end ask user to save it or not. I have also one more function here it is parse. This function change the letter to lowercase and remove the punctuation and spaces.



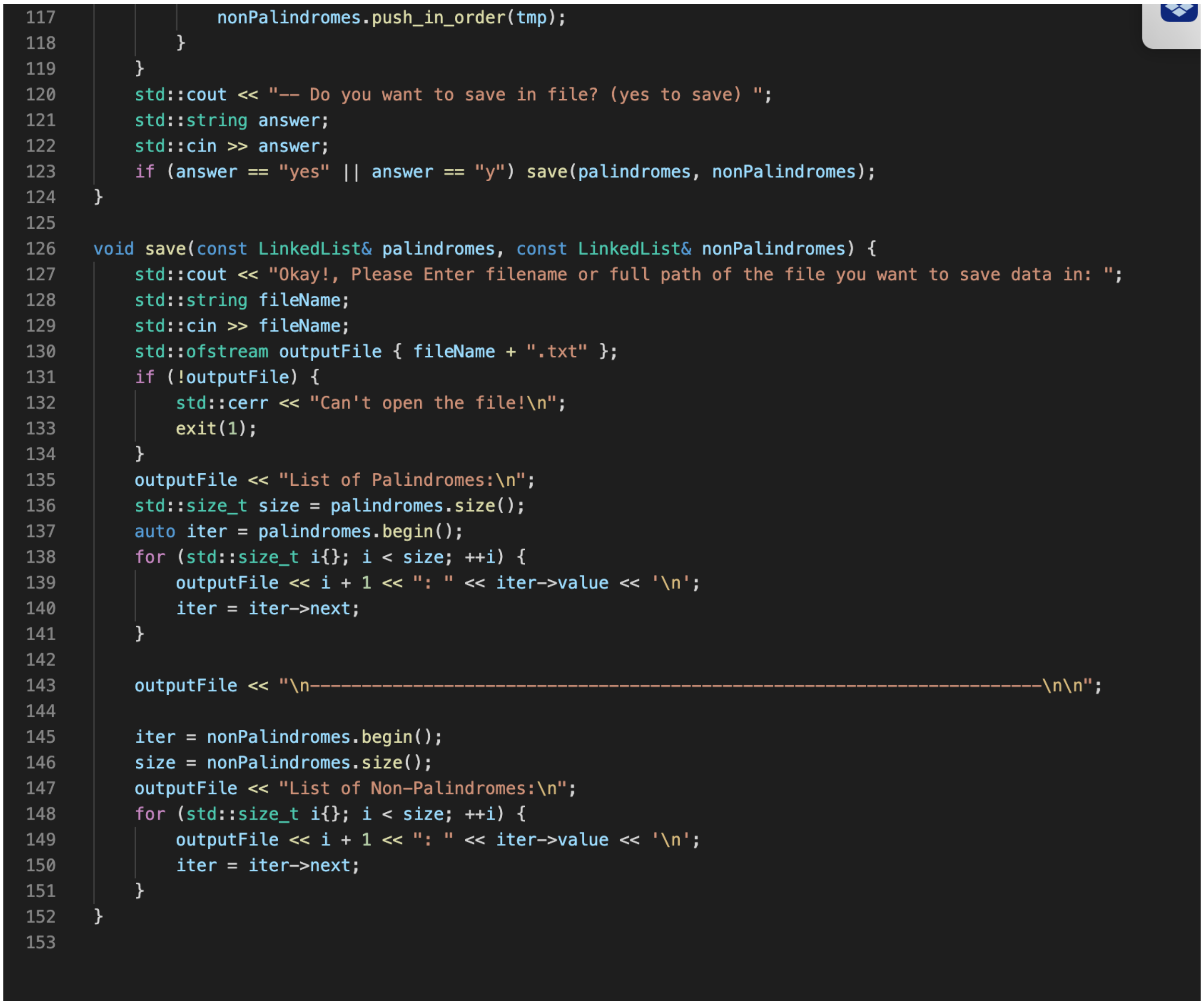
```cpp
void fromConsole() {
    LinkedList palindromes;
    LinkedList nonPalindromes;
    std::cout << "\nOkay! Please Enter your phrases and i'll tell you if they're palindromes or not...\n";
    std::cout << "if you decided to quit just type 'quit' or 'exit'\n";
    std::string input;
    std::cout << "---> ";
    // since i'm going to use std::getline(), i have to ignore the last newline in buffer
    std::cin.ignore(1024, '\n');
    while (std::getline(std::cin, input)) {
        std::string tmp = input;
        if (input == "quit" || input == "exit") break;
        parse(input);
        if (isPalindrome(input)) {
            std::cout << "# Your phrase is palindrome!\n";
            palindromes.push_in_order(tmp);
        }
        else {
            std::cout << "# It's not palindrome\n";
            nonPalindromes.push_in_order(tmp);
        }
        std::cout << "---> ";
    }
    std::cout << "-- Do you want to save in file? (yes to save) ";
    std::string answer;
    std::cin >> answer;
    if (answer == "yes" || answer == "y") save(palindromes, nonPalindromes);
}

void parse(std::string& str) {
    for (std::size_t i{}; i < str.size(); ++i) {
        str[i] = std::tolower(str[i]);
        if (std::ispunct(str[i]) || std::isblank(str[i])) {
            str.erase(str.begin() + i);
            --i;
        }
    }
}

bool isPalindrome(const std::string& str) {
    Stack s;
    Queue q;
    for (char c : str) {
        s.push(c);
        q.push(c);
```

You can see the function “isPalindrome” , I define the two variables s and q. s is stack type and q is queue type. If the s.pop is equal with q.pop and then it is palindrome, if not it is non palindrome. All of the option in the function“fromConsole” is same as the function “fromFile”.



```
bool isPalindrome(const std::string& str) {
    Stack s;
    Queue q;
    for (char c : str) {
        s.push(c);
        q.push(c);
    }
    std::size_t size = s.size();
    for (std::size_t i{}; i < size; ++i) if (s.pop() != q.pop()) return false;
    return true;
}

void fromFile() {
    LinkedList palindromes;
    LinkedList nonPalindromes;
    std::cout << "\nYou decided to read from file...";
    std::cout << "\nPlease give me path to your file: ";
    std::string file;
    std::cin >> file;
    std::ifstream inputFile { file };
    if (!inputFile) {
        std::cerr << "Can't open the file!\n";
        exit(1);
    }
    std::string input;
    // since i'm going to use std::getline() i have to ignore the last newline in buffer
    // std::cin.ignore(1024, '\n') ignores 1024 characters untill newline
    std::cin.ignore(1024, '\n');
    while (std::getline(inputFile, input)) {
        std::string tmp = input;
        parse(input);
        std::cout << "---> " << tmp << '\n';
        if (isPalindrome(input)) {
            std::cout << "# Your phrase is palindrome!\n";
            palindromes.push_in_order(tmp);
        }
        else {
            std::cout << "# It's not palindrome\n";
            nonPalindromes.push_in_order(tmp);
        }
    }
    std::cout << "-- Do you want to save in file? (yes to save) ";
    std::string answer;
    std::cin >> answer;
    if (answer == "yes" || answer == "y") save(palindromes, nonPalindromes);
```

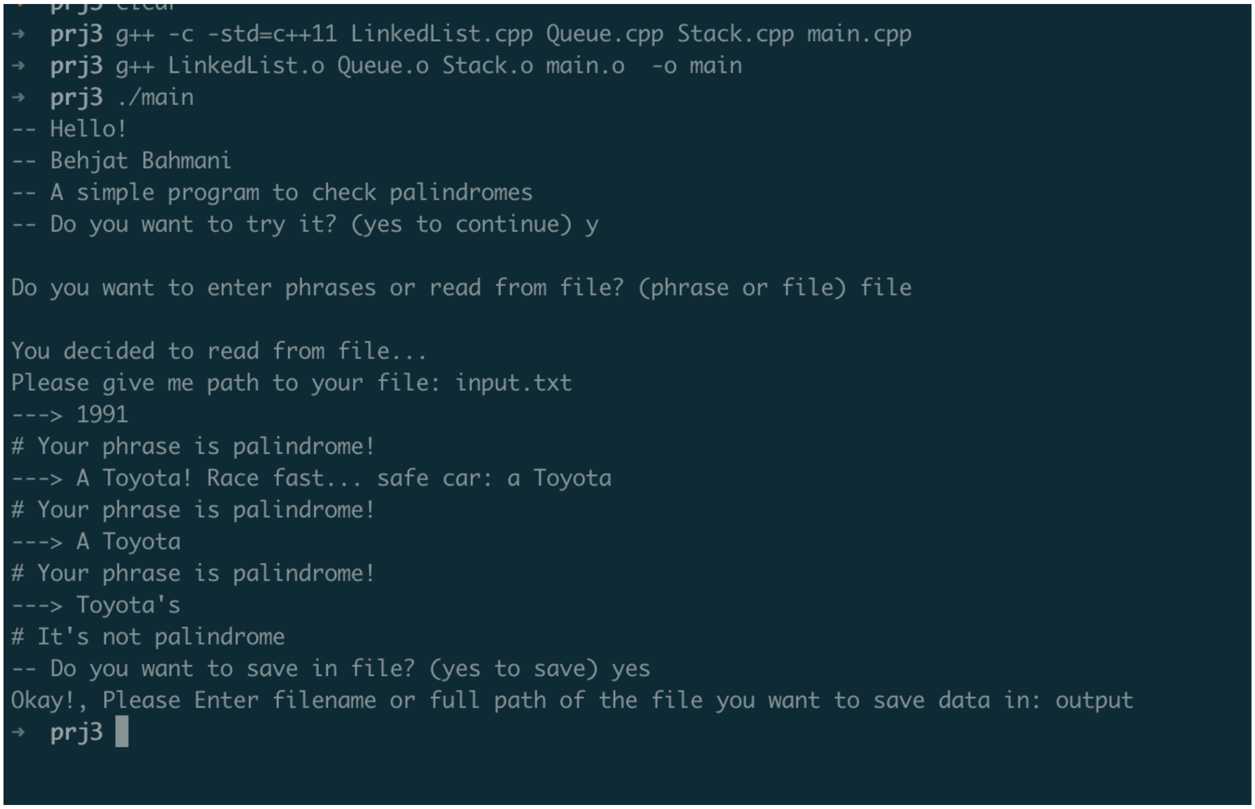
At the end I ask the user if he/she wants to save it. In the function save ask the user to write the filename or full path of the file she/he wants to save data in it. Inside the file I create two part first part is palindromes phrases and the second part is non-palindromes.



```
            nonPalindromes.push_in_order(tmp);
        }
    }
    std::cout << "-- Do you want to save in file? (yes to save) ";
    std::string answer;
    std::cin >> answer;
    if (answer == "yes" || answer == "y") save(palindromes, nonPalindromes);
}

void save(const LinkedList& palindromes, const LinkedList& nonPalindromes) {
    std::cout << "Okay!, Please Enter filename or full path of the file you want to save data in: ";
    std::string fileName;
    std::cin >> fileName;
    std::ofstream outputFile { fileName + ".txt" };
    if (!outputFile) {
        std::cerr << "Can't open the file!\n";
        exit(1);
    }
    outputFile << "List of Palindromes:\n";
    std::size_t size = palindromes.size();
    auto iter = palindromes.begin();
    for (std::size_t i{}; i < size; ++i) {
        outputFile << i + 1 << ": " << iter->value << '\n';
        iter = iter->next;
    }

    outputFile << "\n-----------------------------------------------------------------------\n\n";

    iter = nonPalindromes.begin();
    size = nonPalindromes.size();
    outputFile << "List of Non-Palindromes:\n";
    for (std::size_t i{}; i < size; ++i) {
        outputFile << i + 1 << ": " << iter->value << '\n';
        iter = iter->next;
    }
}

```

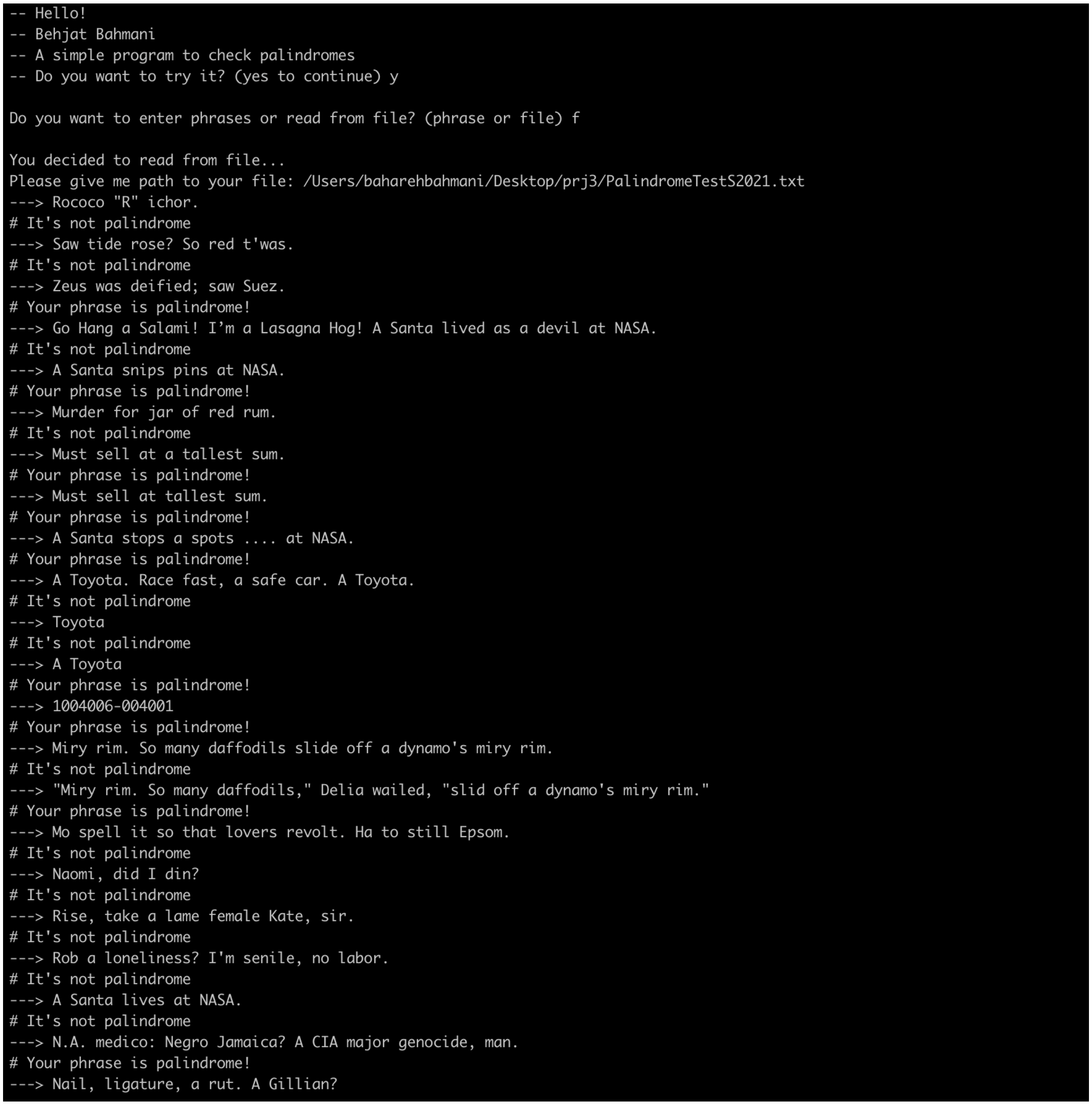
Product:

If user chooses to write a phrase or phrases.



```
➜  prj3 clear
➜  prj3 g++ -c -std=c++11 LinkedList.cpp Queue.cpp Stack.cpp main.cpp
➜  prj3 g++ LinkedList.o Queue.o Stack.o main.o  -o main
➜  prj3 ./main
-- Hello!
-- Behjat Bahmani
-- A simple program to check palindromes
-- Do you want to try it? (yes to continue) y

Do you want to enter phrases or read from file? (phrase or file) file

You decided to read from file...
Please give me path to your file: input.txt
---> 1991
# Your phrase is palindrome!
---> A Toyota! Race fast... safe car: a Toyota
# Your phrase is palindrome!
---> A Toyota
# Your phrase is palindrome!
---> Toyota's
# It's not palindrome
-- Do you want to save in file? (yes to save) yes
Okay!, Please Enter filename or full path of the file you want to save data in: output
➜  prj3 
```

If user chooses a file.



```
-- Hello!
-- Behjat Bahmani
-- A simple program to check palindromes
-- Do you want to try it? (yes to continue) y

Do you want to enter phrases or read from file? (phrase or file) f

You decided to read from file...
Please give me path to your file: /Users/baharehbahmani/Desktop/prj3/PalindromeTestS2021.txt
---> Rococo "R" ichor.
# It's not palindrome
---> Saw tide rose? So red t'was.
# It's not palindrome
---> Zeus was deified; saw Suez.
# Your phrase is palindrome!
---> Go Hang a Salami! I’m a Lasagna Hog! A Santa lived as a devil at NASA.
# It's not palindrome
---> A Santa snips pins at NASA.
# Your phrase is palindrome!
---> Murder for jar of red rum.
# It's not palindrome
---> Must sell at a tallest sum.
# Your phrase is palindrome!
---> Must sell at tallest sum.
# Your phrase is palindrome!
---> A Santa stops a spots .... at NASA.
# Your phrase is palindrome!
---> A Toyota. Race fast, a safe car. A Toyota.
# It's not palindrome
---> Toyota
# It's not palindrome
---> A Toyota
# Your phrase is palindrome!
---> 1004006-004001
# Your phrase is palindrome!
---> Miry rim. So many daffodils slide off a dynamo's miry rim.
# It's not palindrome
---> "Miry rim. So many daffodils," Delia wailed, "slid off a dynamo's miry rim."
# Your phrase is palindrome!
---> Mo spell it so that lovers revolt. Ha to still Epsom.
# It's not palindrome
---> Naomi, did I din?
# It's not palindrome
---> Rise, take a lame female Kate, sir.
# It's not palindrome
---> Rob a loneliness? I'm senile, no labor.
# It's not palindrome
---> A Santa lives at NASA.
# It's not palindrome
---> N.A. medico: Negro Jamaica? A CIA major genocide, man.
# Your phrase is palindrome!
---> Nail, ligature, a rut. A Gillian?
```

Pitfalls:

I didn’t install valgrind because I had some errores. I tried to replace the head url and then run it but it didn’t work.

Possible improvements:

I am not sure but maybe it can be possible without exiting from program we chose some pharaces and also after that we choose the file and save all of them together.