

**Preparatory data Structure (CSCI 591)**



**Project - II**

**Recursive Functions – Three small programs**

**January 28, 2020**

**St. Cloud state university**

**Department of Computer Science**

Submitted By: Taddese Erba

# **Design Document**

## **Introduction**

A recursive function is a function that calls itself. By calling itself, a recursive function executes a block of code repeatedly until it hits the smallest possible result(s) also known as the base. Recursive code is important because of its small code size and easiness to write. Once it is written, it can also be used over and over. A recursive function is not always the desired method for solving a problem due to its performance issues. It often takes a long time to compute and return an output than a loop program.

## **Data Structure**

The program uses three data structures, the class FileStream, class Palindrome, and the function exp(). This class FileStream holds the character input from a .dat file. This class hosts two arrays and three functions as its members. The array char array[MAX] is used to hold the characters read in from the file. The second array, char a[MAX], is used to hold the array of characters while we search the array for the presence of a character.

The second class is the class Palindrome. This class is used to recursively check whether a string entered by the user is a palindrome or not. The class Palindrome has three character type arrays and five-member functions. The array charArr[MAXLEN] holds the characters entered by the user. Another array, arrayToLower[MAXLEN], is used to hold the characters converted to lower case. The whiteSpaceRemoved[MAXLEN] array holds the characters without whitespaces. All the arrays are character data types with their size set to a maximum value.

## **Functions**

There are three functions in the class FileStream and five functions used in the class Palindrome. The function void readFile(char [], int) is used to read the characters from the file one at a time and places these characters in the array char array[MAX]. It takes two arguments and returns nothing. The second function, int getnumOfChars(int), is used to count the number of characters read from the file. It takes in integer type variable and returns an integer to the main function. The last function in this class is the bool present(char [], int, int, char) function that is used to lookup the presence of the character entered by the user. This function takes one character type array, two integer type variables and one character type variable. If it successfully finds or fails to find the character entered by the user, it returns a Boolean result to the main function.

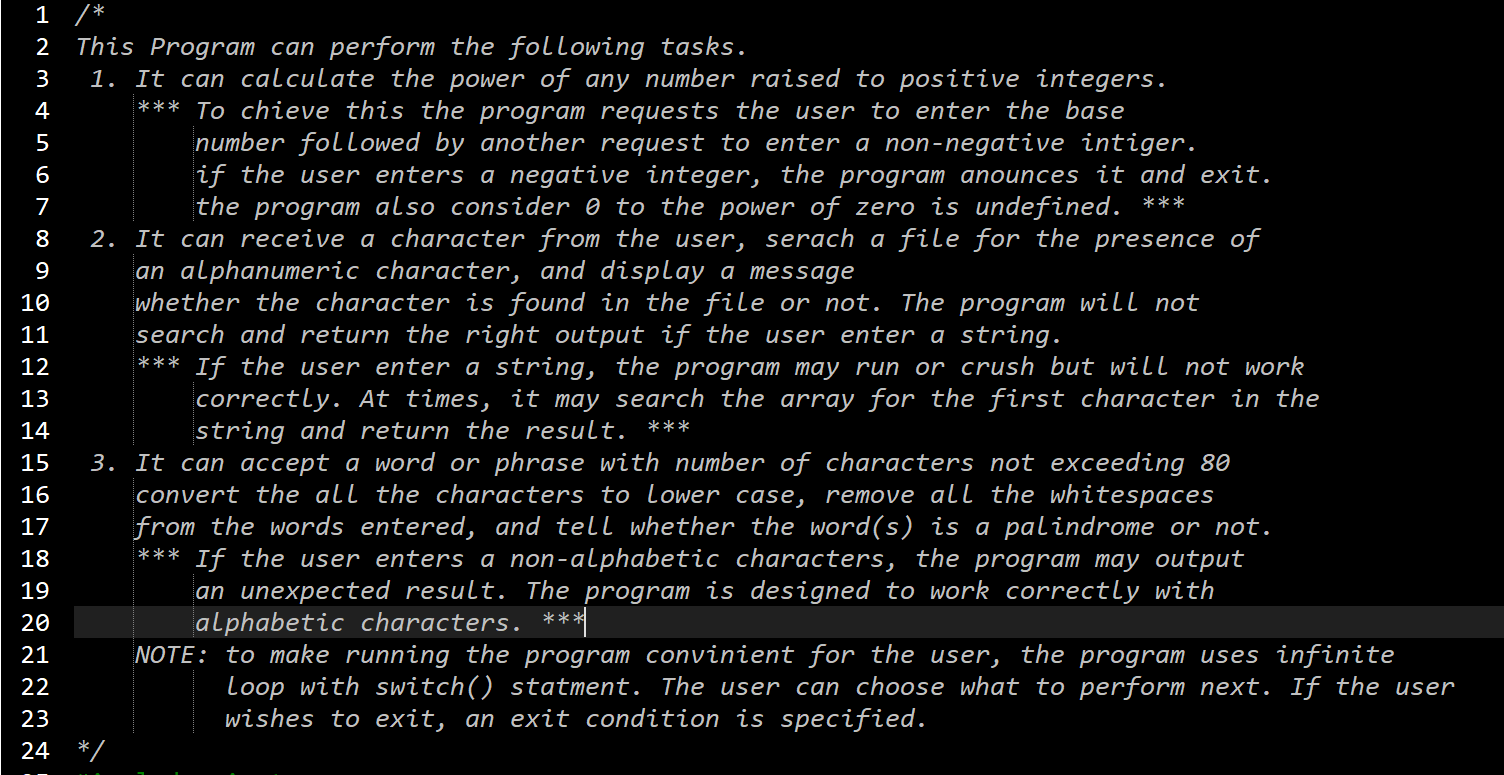
The void getPalindrome (char [], int) function is used to get the stream of characters from the user. This function takes the charArr[] array and another character variable as its arguments and returns no value to the main function. Furthermore, this function checks if the user entered the right size of characters and displays a message to the user accordingly. The function void toLowerCase(char [], int) takes two arguments, the array charArr[] and the array size int. It is used to convert the characters from upper to lower case and has no return value. The third function, void removeWhiteSpaces(char [], int), is similar in its structure to the other mentioned functions. Its main purpose is to remove all the whitespaces from the character arrays. The void reversePalindrome (char [], int) function is used to write the characters in their reverse order (from last to first). Its structure is similar to the above functions and it returns no value. The last function in this program, the bool compare (char [], char [], int, int), is used to compare the original characters entered by the user without the whitespaces to the reversed characters. It has four arguments, charArr[], arrayReversed [], len, and len. The two len are the sizes of the two arrays. This function returns a boolean true or false.

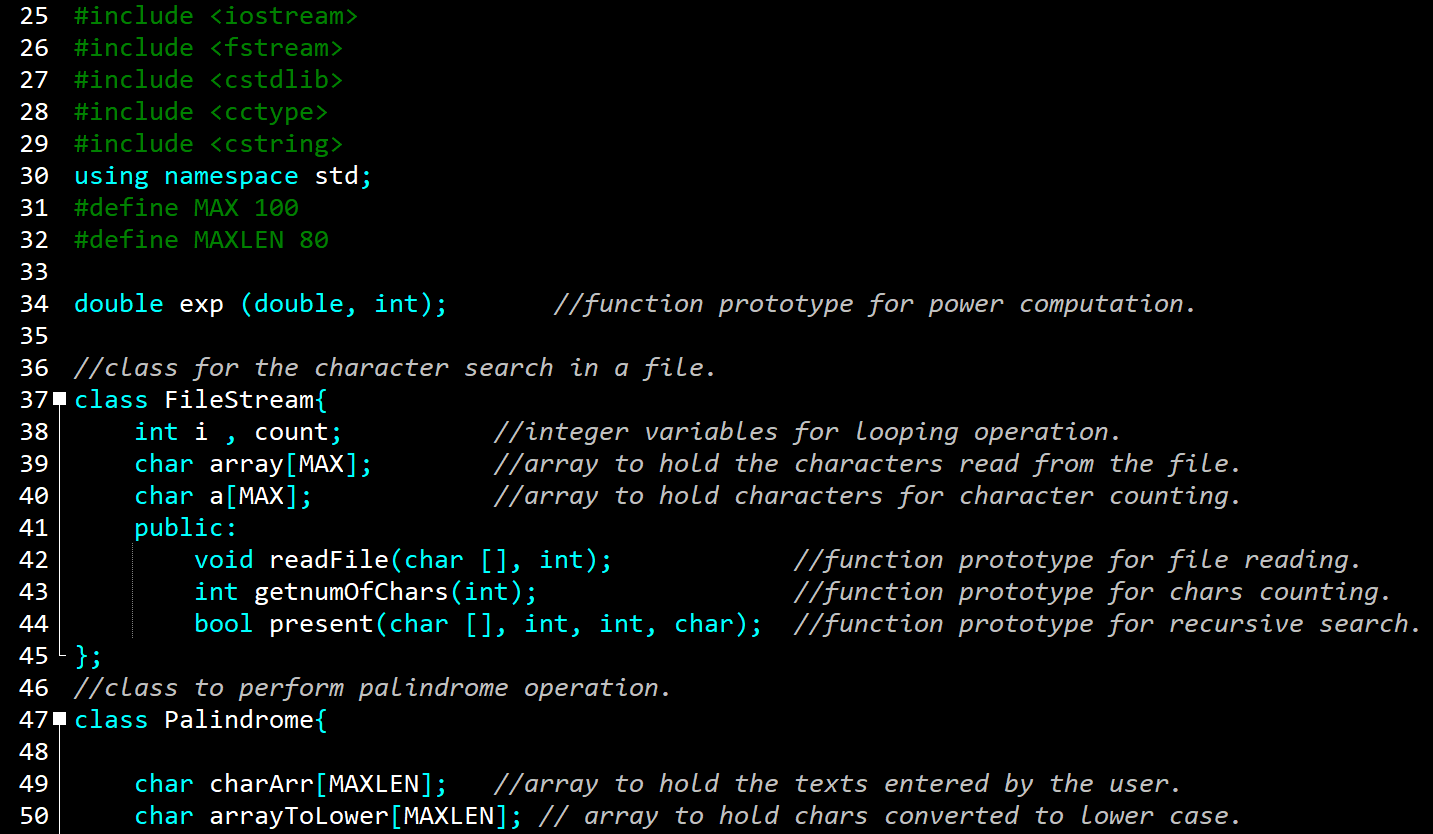
The last function that is used in this program without a class is the double exp (double, int) function. This function performs the computation of exponential (power) of any number raised to positive integers. The function is capable of accepting zero and all positive integers as the exponent. The base can be any real number including zero. This program assumes zero to the power of zero is undefined and returns values accordingly.

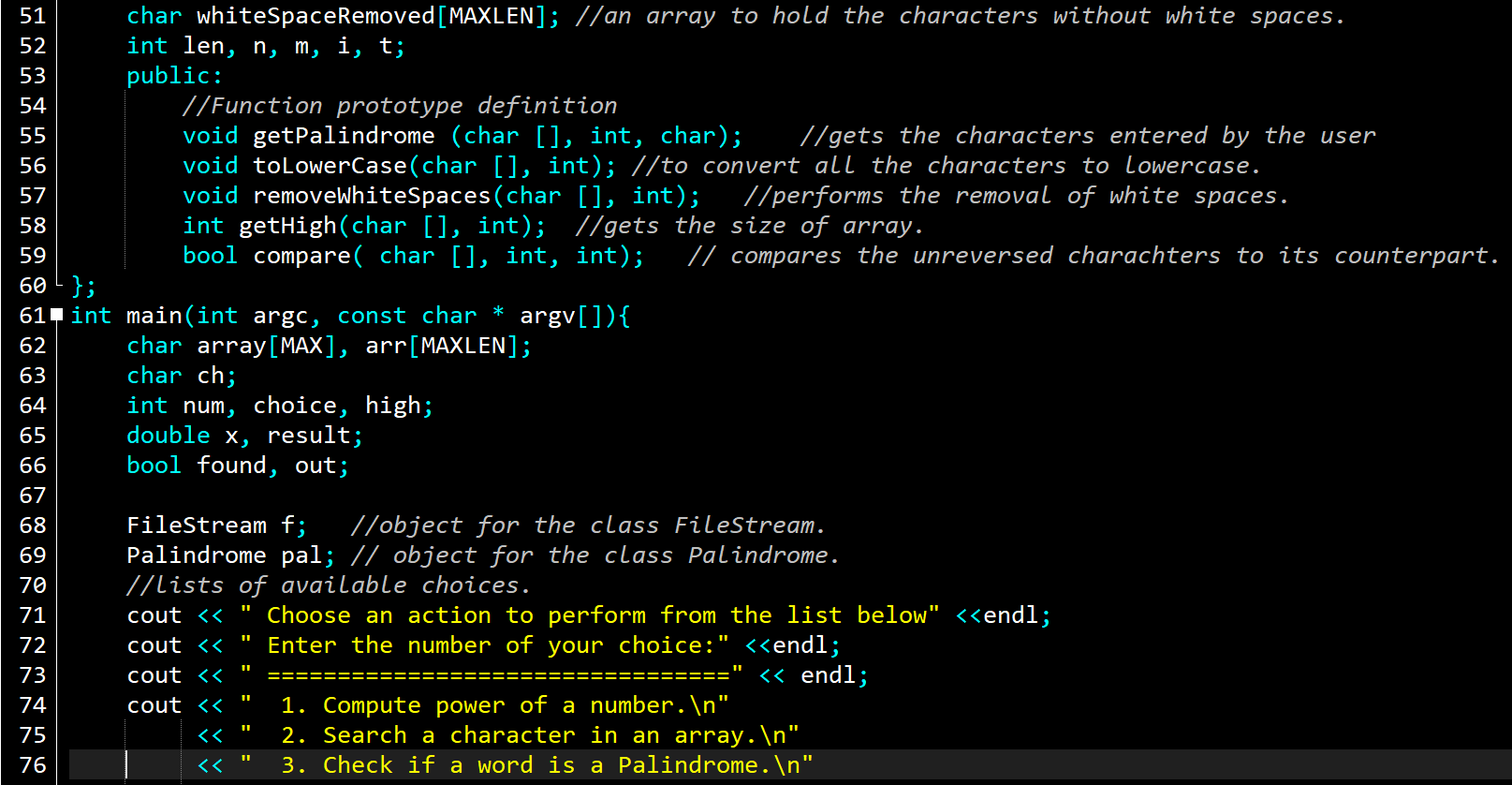
## **The Main Program**

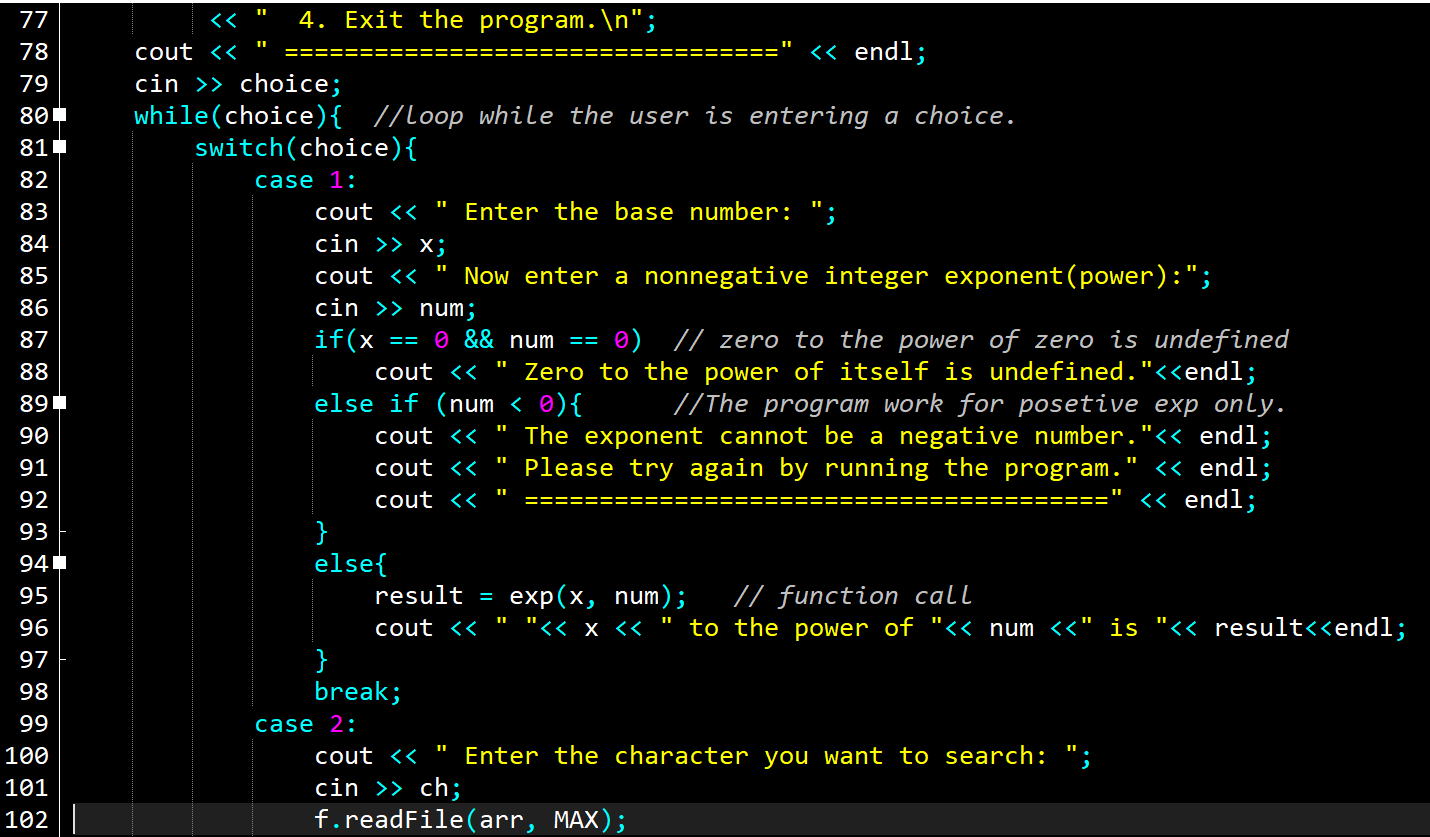
The main program initializes the class Palindrome and class FileStream and calls their respective member functions as necessary. The main function uses a switch () statement inside a forever while () loop to allow the user to perform different computations. There is also an exit condition if the user wants to quit running the program. While it calls and initializes the individual functions, the main will also display a series of messages to aware the user about what is going on.

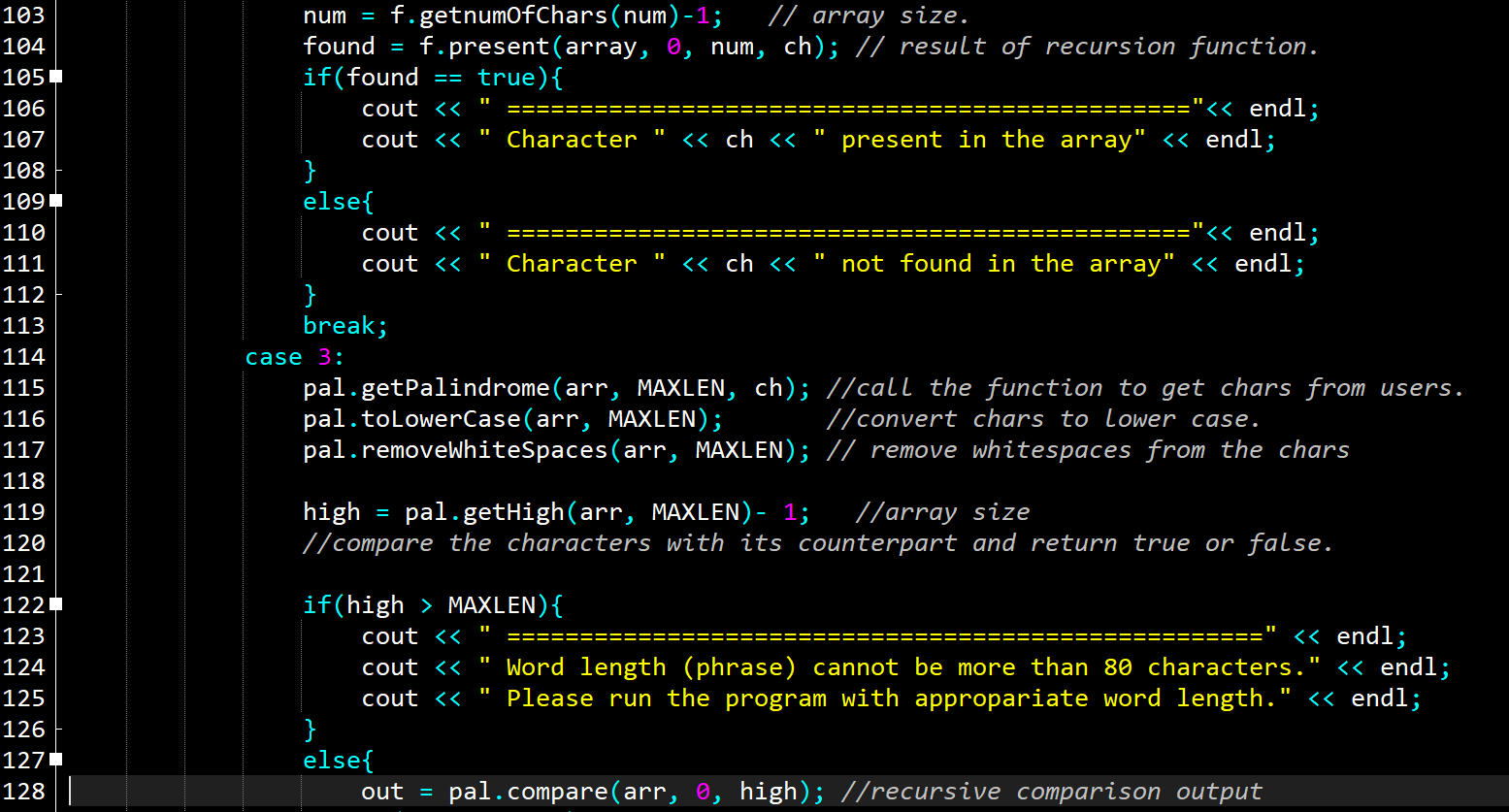
# **Code listing**

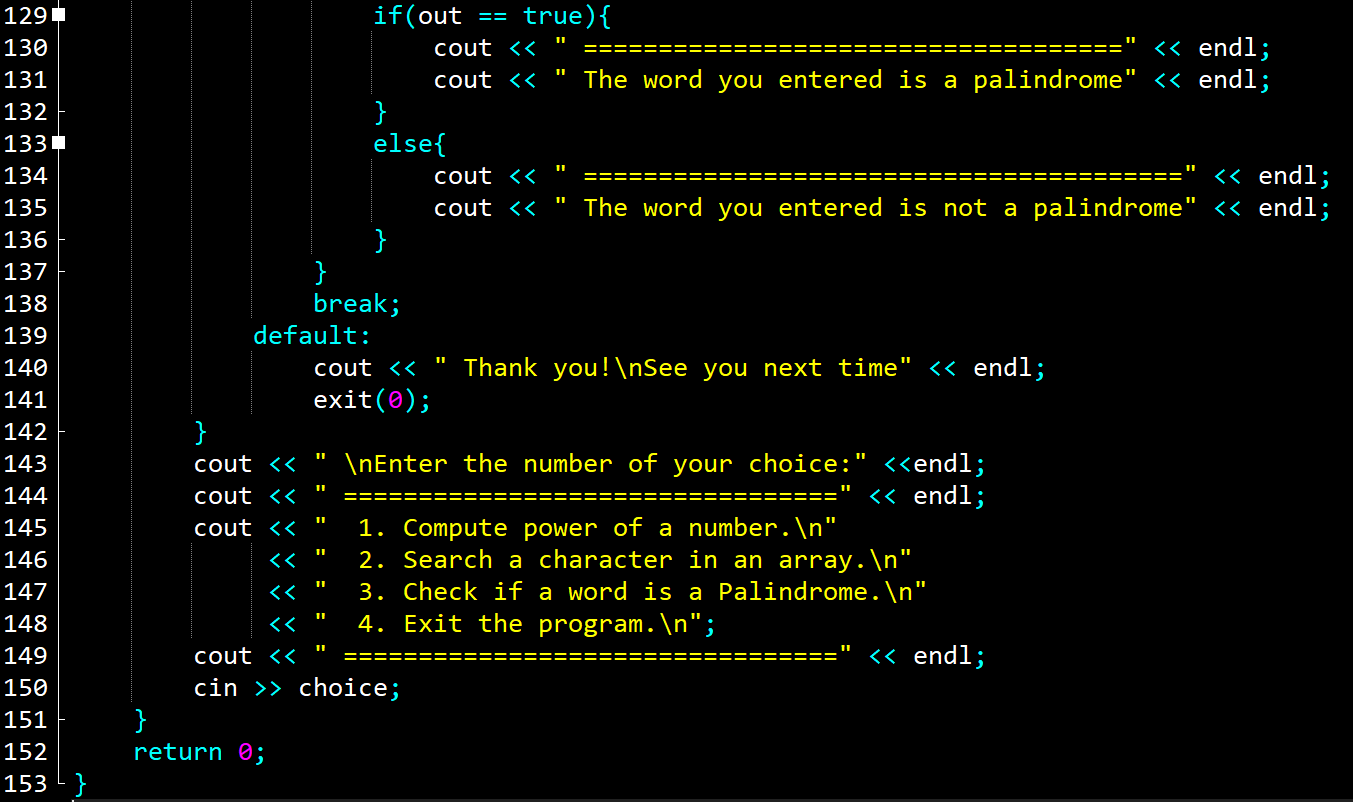


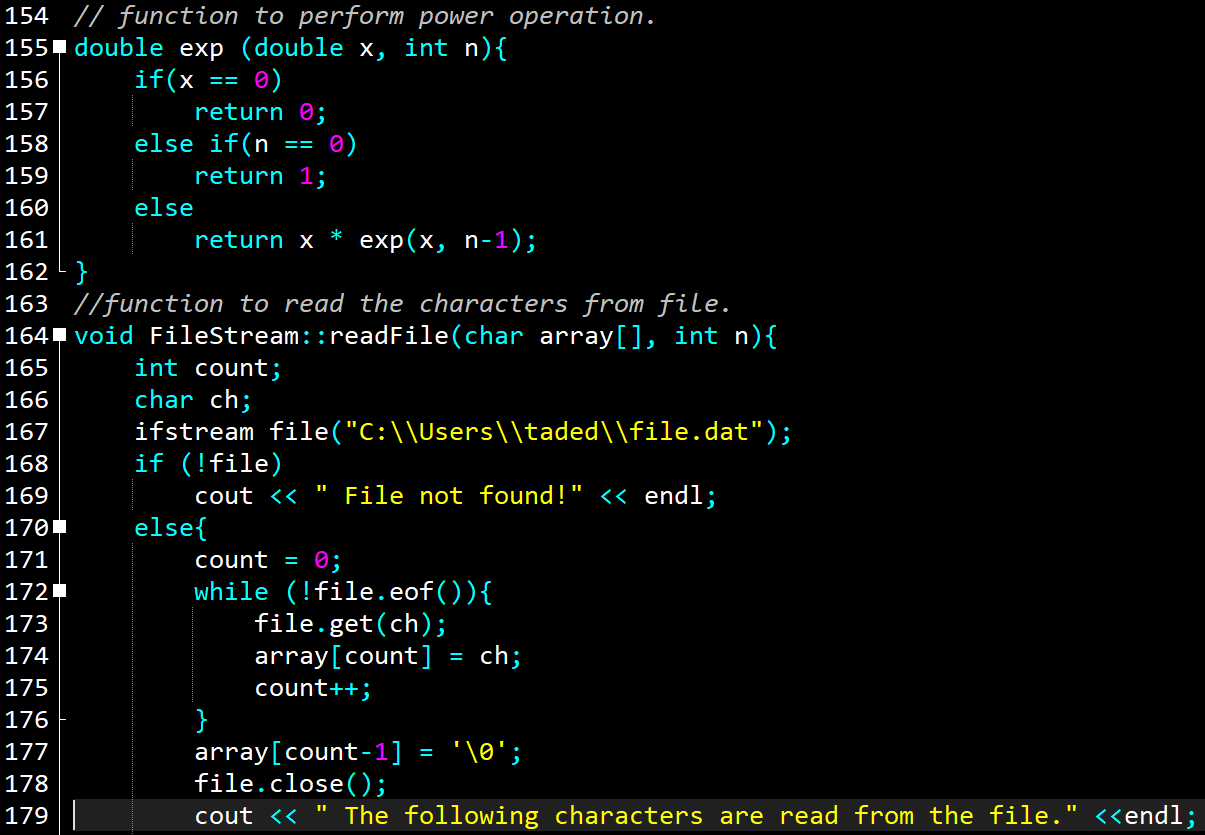


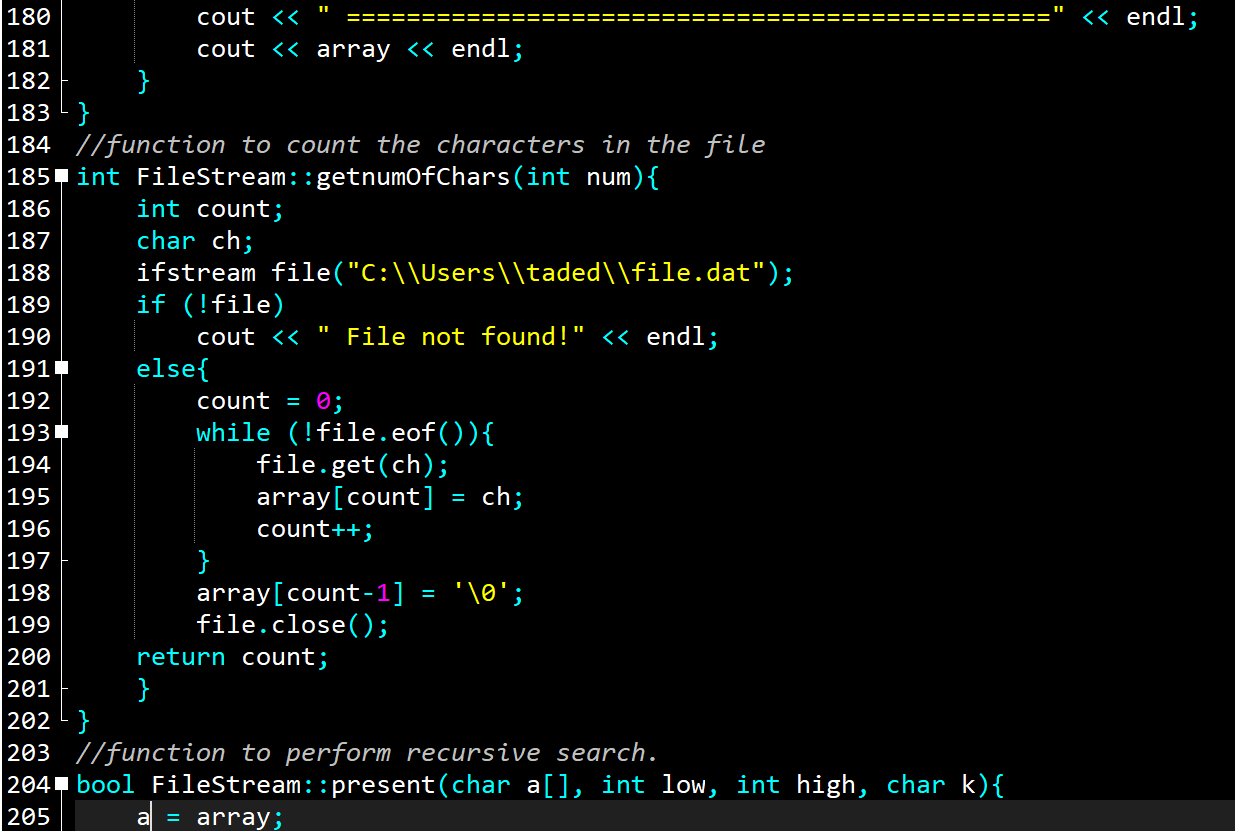


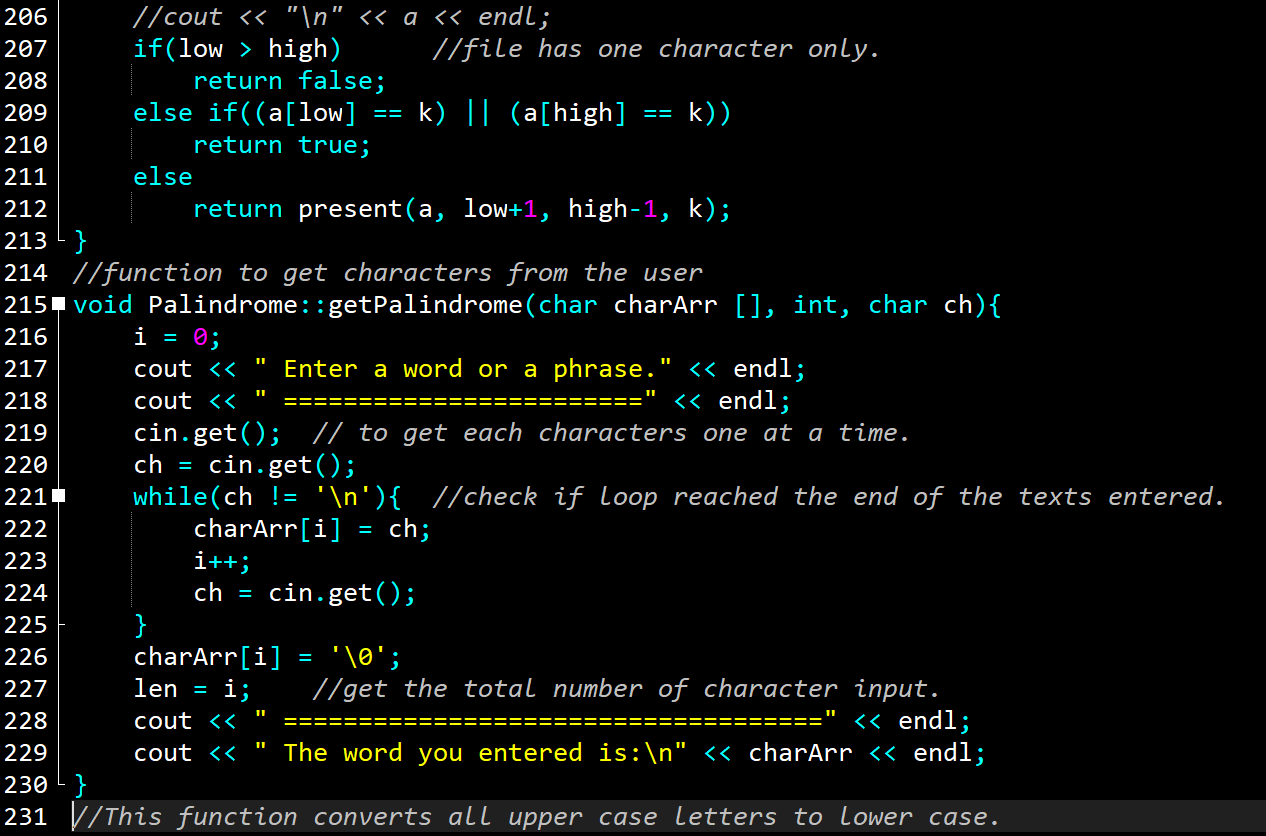


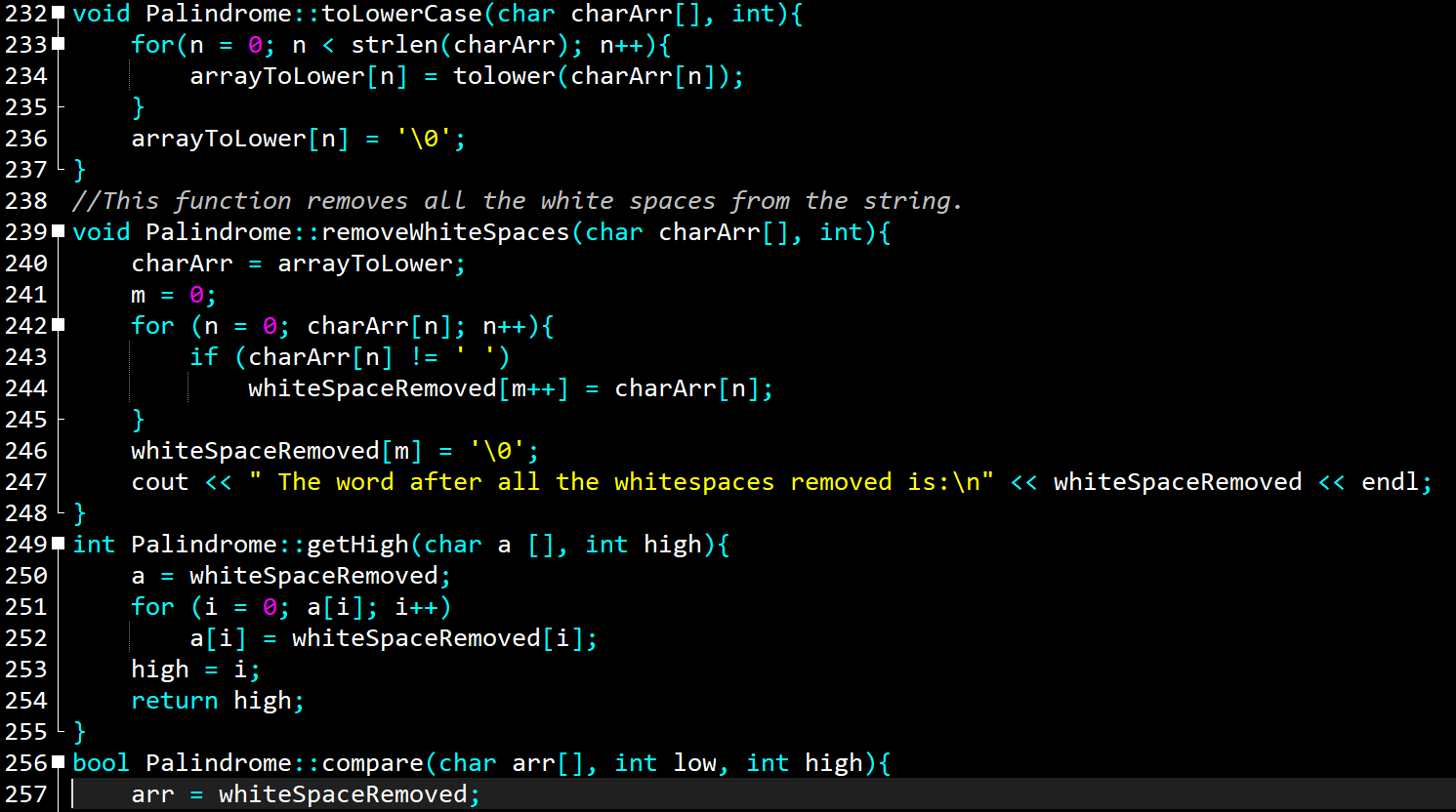


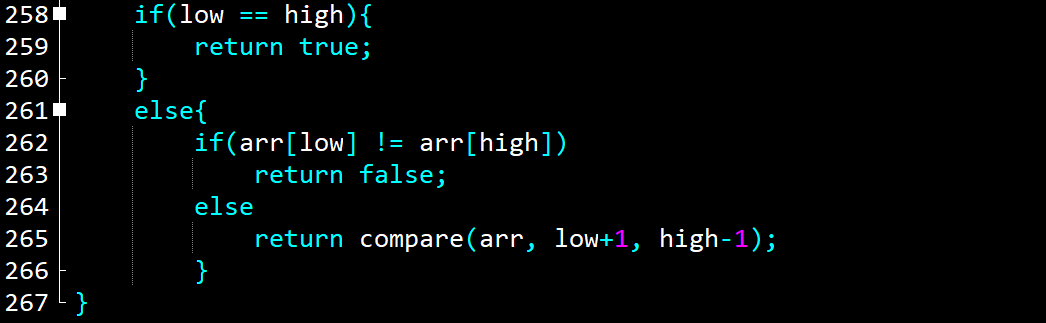






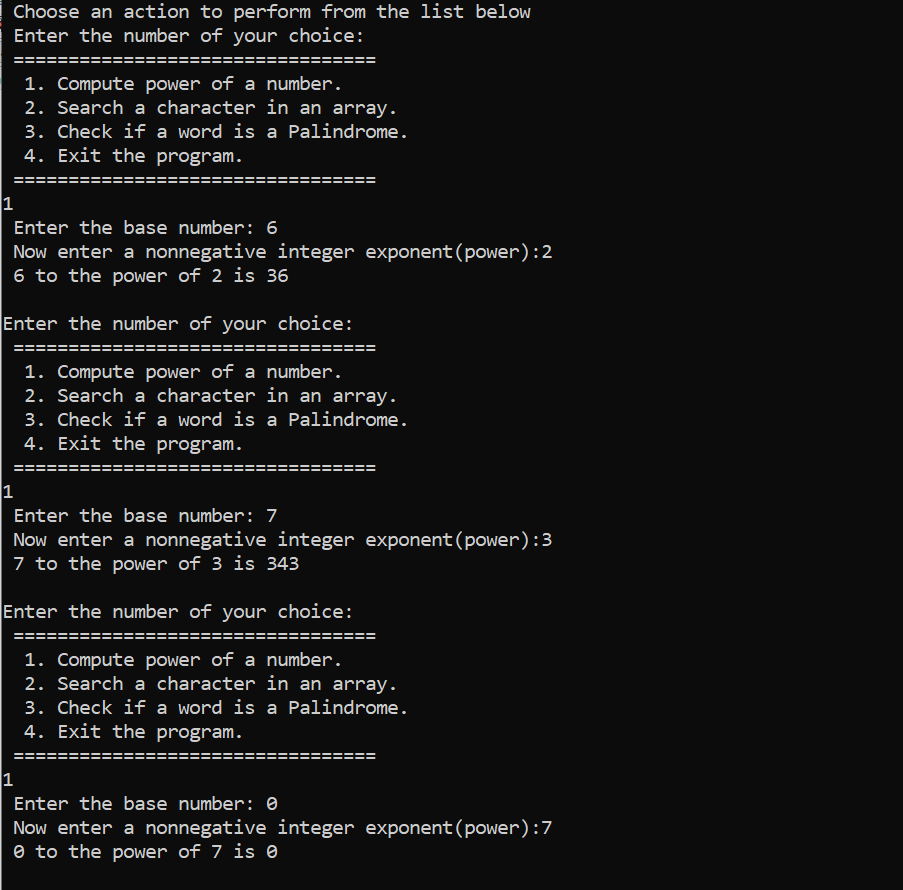


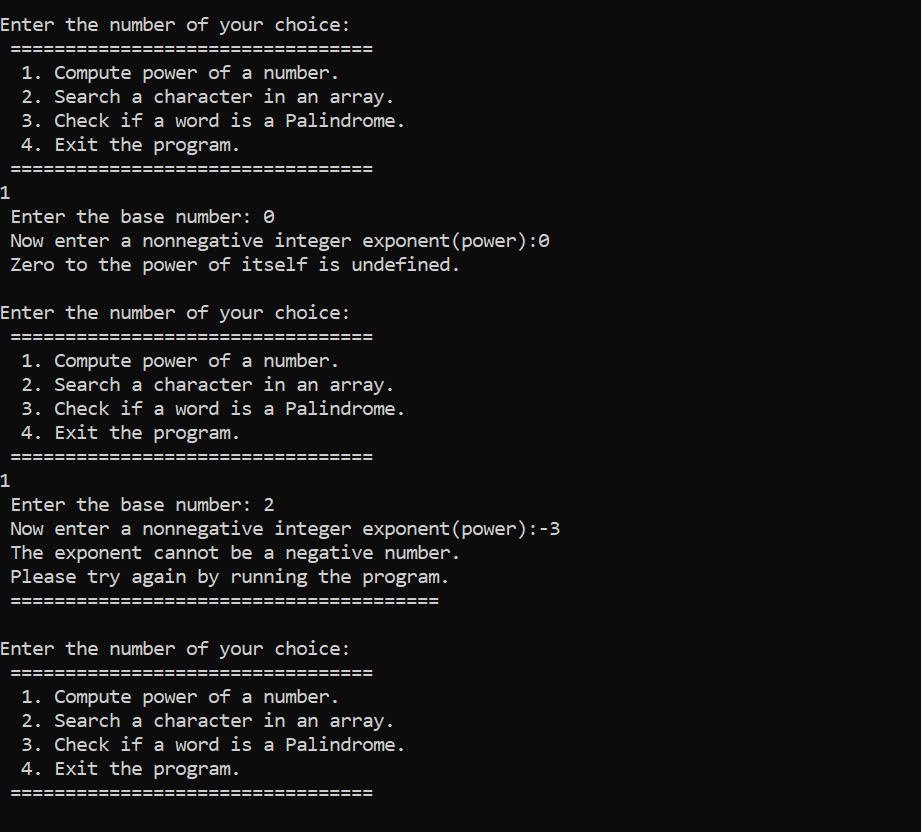




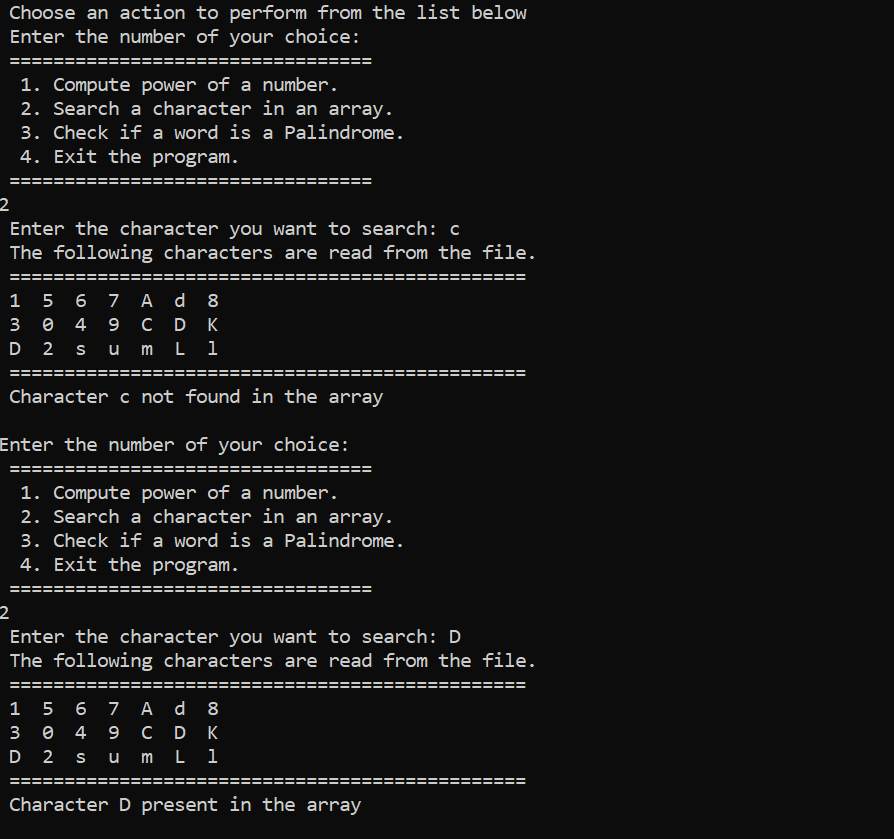
# **Test Results**

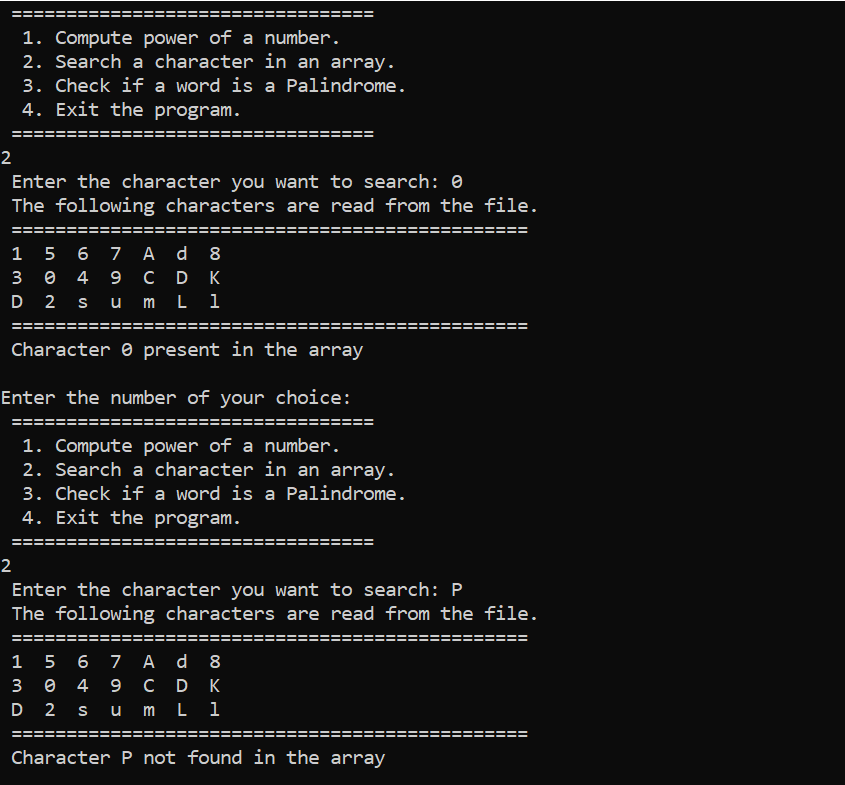
## **Tests result for exponential computation.**





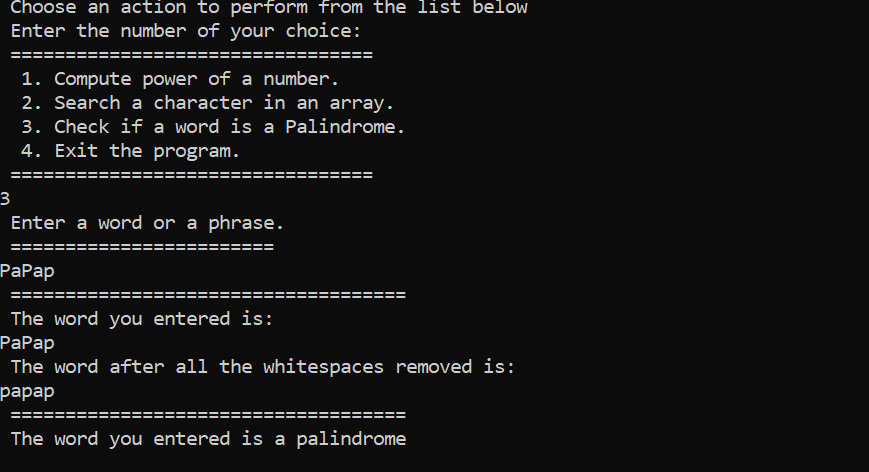
## **Tests result for character search in a file.**



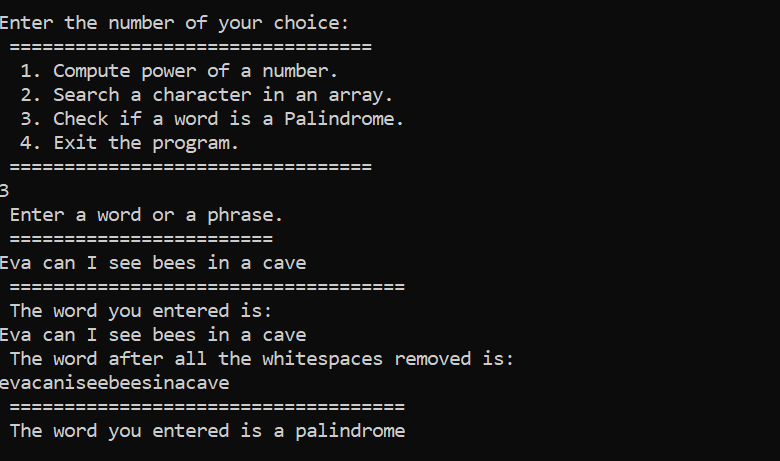


## **Tests with a palindrome word**

### **Short (single) word**

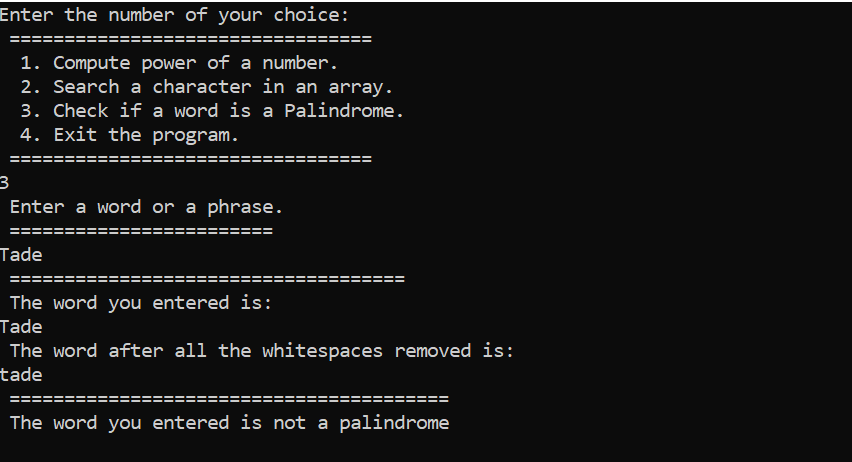


### Long word

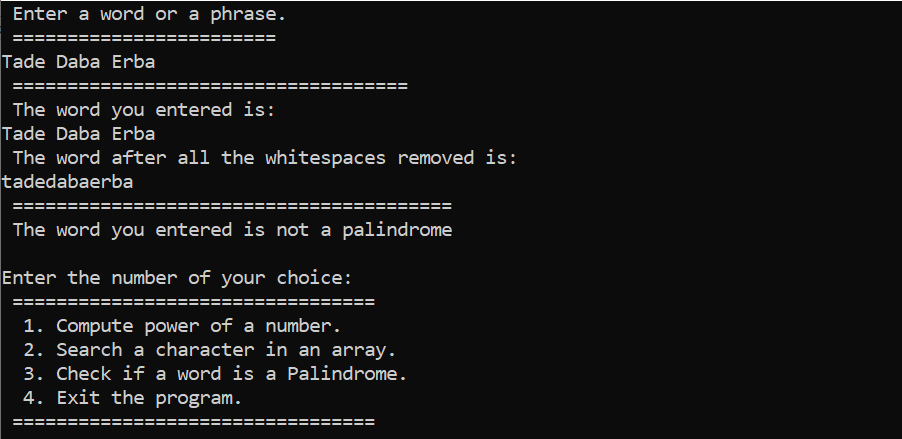


## **Tests with a nonpalindromic word**

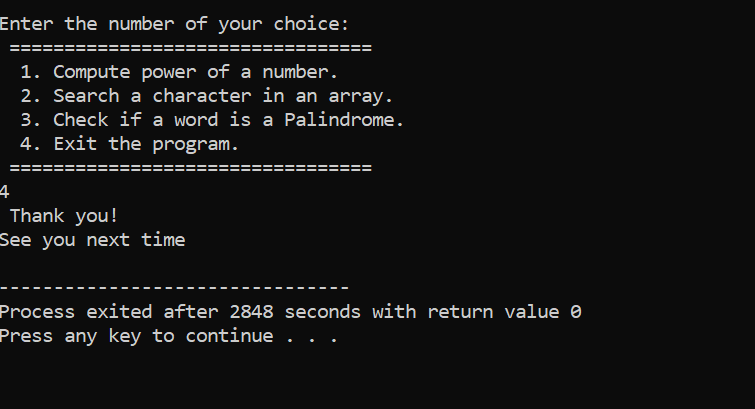
### **Short (single) word**



### Long word



## **Tests for exit condition.**



# **User document**

This program consists of three subprograms that perform different tasks. You can do the following by using the program.

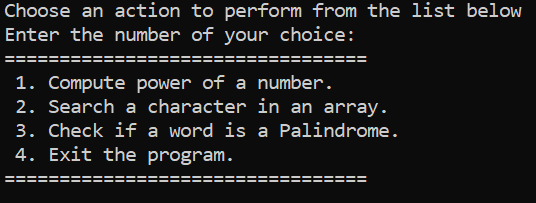
1. Compute the power of any number to a positive integer number.
2. You can search for a character in a file saved on your hard drive.
3. You can check if a word is a palindrome or not by entering a word from the terminal.

In order to run the program, you must perform the following steps.

* The program name is Exponential.cpp. on the terminal enter the following command to compile and run the program.

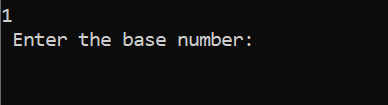
g++ -o Exponential Exponential.cpp

* The program will compile and open the following:

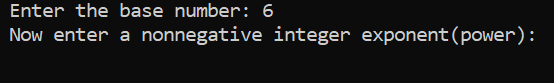


* Make a choice from the displayed menu. For example to compute exponential, type 1 and hit enter.

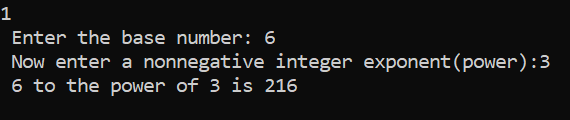
Then the program will ask you to enter a number as shown below.



* Enter a number. The program will display the following.



* Now enter only a positive integer. The program doesn't work with any other number.



* Finally, the program will display the result as shown above.

The steps to perform the rest of the tasks is similar except you will require to type in the word you want to check for a palindrome and you also need to save the file you want to search as file.dat at the following location.

C:\Users\taded\file.date

# **Summery**

This program is the collection of three stand-alone programs that can perform independently. The double exp(int) function exists to perform power computation. Its argument is an integer type indication it can take only an integer and can return a double or an integer. The classes FileStream and Palindrome are also stand-alone programs that can operate on character type arrays. The main function uses while() loop in conjunction with the switch() statement to create convenience for the user to perform the three independent tasks without the need to restart and run the program time and again.

This program can be used to perform mathematical computation (power computations), to implement search algorithms, and to control the size and nature of the characters entered by the user into some character limiter fields such as username and password fields. The program can be further improved by modifying the functions. For instance, the character search program can be improved to make a string search where the string size can be a character or the whole file or paragraph. Similarly, the palindrome program can be improved in a way it can perform interesting tasks such us limiting the character type and size entered by the user into some character accepting fields such as usernames and passwords. By completing this project, I have gained enough knowledge and experience to work with recursive functions and their implementation. One interesting thing about recursive functions that I came across as I complete this project is their simplicity in writing in programming code.