

Student: Matthew Smyth

Project Due Date: 09/07/2020

Algorithm Steps:

Step 0: outFile open argv[1] to write

Step 1: displayRules (outFile)

Step 2: askPasswd (outFile)

password screen input from the user

passwordLength <-- the length of password (use strlen)

step 3: repeat step 2 if the length of password is NOT within the range of 8 - 32

print error message

step 4: i 0

step 5: index checkCharType(password[i]) // make sure the index is within 0 - 4

charCount[index] ++

step 6: i++

step 7: repeat step 5 to step 6 until the last password character is checked.

step 8: validYesNo checkRules ()

if validYesNo is not good (!= 1)

call displayFail(outFile)

step 9: repeat step 1 to step 8 if validYesNo is 0

step 10: display and ask user to re-type his/her password; also write to outFile

step 11: secondPassword from the user

step 12: matchYesNo matching (password, secondPassword)

step 13: if matchYesNo is no good (== 0)

displayMatchFail(outFile)

step 14: repeat step 1 to step 13 until matchYesNo == 1

step 15: call displaySuccess (outFile)

```
#include <iostream>
```

```
#include <fstream>
```

```
using namespace std;
```

```
class passwordChecker{
```

```
    public:
```

```
    string pw;
```

```
    string secondPW;
```

```
    int pwLength;
```

```
    int charCount[5] = {0};
```

```
public:
```

```
    passwordChecker(){
```

```
        this->pw = "";
```

```
        this->pwLength = 0;
```

```
    }
```

```
    void displayRules(ofstream & outFile){
```

```
        cout << "Password rules:\n" << "Password length: 8-32 characters\n" <<
```

```
"At least one numerical, i.e., 0, 1, 2,.. \n" << "At least one upper case letter, i.e., A, B, C, ... \n";
```

```
        cout << "At least one lower case letter, i.e., a, b, c, ... \n";
```

```
        cout << "At least one of the special characters, but it must be within the  
set:{ # $ * ( ) % & ^} a total of eight (8) special characters. no other special characters are  
allowed.\n";
```

```
        outFile << "Password rules:\n" << "Password length: 8-32 characters\n"
```

```
<< "At least one numerical, i.e., 0, 1, 2,.. \n" << "At least one upper case letter, i.e., A, B, C, ... \n";
```

```
        outFile << "At least one lower case letter, i.e., a, b, c, ... \n";
```

```
        outFile << "At least one of the special characters, but it must be within the  
set:{ # $ * ( ) % & ^} a total of eight (8) special characters. no other special characters are  
allowed.\n";
```

```
    }
```

```
    void askPassword(ofstream & outFile){
```

```
        while(this->pw.length() < 8 || this->pw.length() > 32){
```

```
            cout << "Please enter password: ";
```

```
            outFile << "Please enter password: ";
```

```
            cin >> this->pw;
```

```
            this->pwLength = this->pw.length();
```

```
            if(this->pw.length() < 8 || this->pw.length() > 32){
```

```
                cout << "Incorrect password length";
```

```
                outFile << "Incorrect password length";
```

```
            }
```

```

        }
        cout << pw;
        outFile << pw;
    }

    void displaySuccess(ofstream & outFile){
        cout << "Your password will be updated in a few minutes!";
        outFile << "Your password will be updated in a few minutes!";
    }

    void displayFail(ofstream & outFile){
        cout << "Your password failed one or more password rules";
        outFile << "Your password failed one or more password rules";
    }

    void displayMatchFail(ofstream & outFile){
        cout << "Match fail";
        outFile << "Match fail";
    }

    int checkCharType(char ch){
        int num = int(ch);
        if(num >= 48 && num <= 57){
            return 1;
        } else if(num >= 97 && num <= 122){
            return 2;
        } else if (num >= 65 && num <= 90){
            return 3;
        } else if (num == 35 || num == 36 || num == 42 || num == 40 || num == 41
|| num == 37 || num == 38 || num == 94){
            return 4;
        } else{
            return 0;
        }
    }

    bool checkRules(int charC[]){
        for(int i = 1; i<5; i++){
            if(charC[i] == 0){
                return 0;
            }
        }
        if(charC[0] > 0){

```

```

        return 0;
    }
    return 1;
}

bool matching(string s1, string s2){
    int greater = max(s1.length(), s2.length());
    for(int i = 0; i < greater; i++){
        if(s1[i] != s2[i]){
            return 0;
        }
    }
    return 1;
}

};

int main(int argc, char** argv) {
    int validYesNo;
    int matchYesNo;

    string outputName = argv[1];
    ofstream outFile;
    outFile.open(outputName);

    passwordChecker *check = new passwordChecker();

    while(validYesNo != 1){
        check->displayRules(outFile);
        check->askPassword(outFile);

        for(int i = 0; i < check->pw.length(); i++){
            int index = check->checkCharType(check->pw[i]);
            check->charCount[index]++;
        }

        validYesNo = check->checkRules(check->charCount);
        if(validYesNo != 1){
            check->displayFail(outFile);
        }
    }

    while(matchYesNo != 1){

```

```

        cout << "\nReenter Password: ";
        outFile << "\nReenter Password: ";

        string again;
        cin >> again;
        check->secondPW = again;
        cout << again;
        outFile << again;
        matchYesNo = check->matching(check->pw, check->secondPW);
        if(matchYesNo != 1){
            check->displayMatchFail(outFile);
        }
    }

    check->displaySuccess(outFile);

    return 0;
}

```

Output file:

Password rules:

Password length: 8-32 characters

At least one numerical, i.e., 0, 1, 2,..

At least one upper case letter, i.e., A, B, C, ...

At least one lower case letter, i.e., a, b, c, ...

At least one of the special characters, but it must be within the set:{ # \$ \* ( ) % & ^} a total of eight (8) special characters. no other special characters are allowed.

Please enter password: Incorrect password lengthPlease enter password: Password1#

Reenter Password: nopeMatch fail

Reenter Password: Password1#Your password will be updated in a few minutes!

Console output:

C:\Users\supab\Desktop\323 design analysis algorithms  
 projects\SmythM\_Project2\_CPP>SmythM\_Project2\_CPP.exe output.txt

Password rules:

Password length: 8-32 characters

At least one numerical, i.e., 0, 1, 2,...

At least one upper case letter, i.e., A, B, C, ...

At least one lower case letter, i.e., a, b, c, ...

At least one of the special characters, but it must be within the set: { # \$ \* ( ) % & ^ } a total of eight (8) special characters. no other special characters are allowed.

Please enter password: fail

Incorrect password lengthPlease enter password: Password1#

Password1#

Reenter Password: nope

nopeMatch fail

Reenter Password: Password1#

Password1#Your password will be updated in a few minutes!