21CSCI01P

Introduction to Computing

Lab (4)

This tutorial covers

- **Tutorial:**
    1. **Files**
- **Problems from 17 to 19 in the lecture slides.**
- **Chapter 6 in the textbook, 8th edition.**

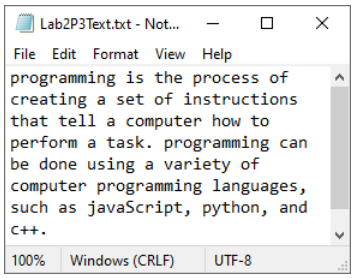
Problem (1): Extracting data from a file to another file:

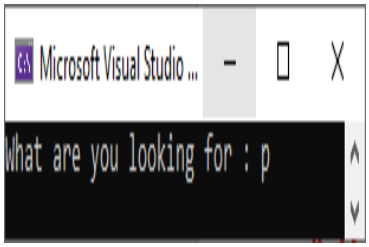
1. Consider you have a file named “Lab4Text.txt”, that contains the following text:

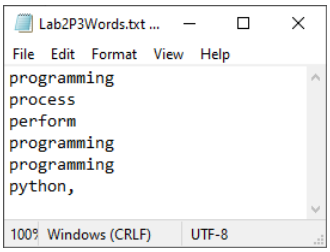
programming is the process of creating a set of instructions that tell a computer how to perform a task. programming can be done using a variety of computer programming languages, such as javaScript, python, and c++.

2. Take a character as input from the user. Use this character to search for and get all words in the file that begin with this character.
3. Saves the extracted words in a file named “Lab4Words.txt”.

For example:







Solution (1)

```
#include<iostream>
#include<fstream>
#include<string>
using namespace std;
int main() {
	char c;
	string word;
	fstream fin, fout;
	fin.open("Lab2P3Text.txt", ios::in);
	fout.open("Lab2P3Words.txt", ios::out);
	cout << "What are you looking for : "; cin >> c;
	while (fin>>word) {
		if (word.at(0) == c) {
			fout << word << endl;
		}
	}
	fin.close();
	fout.close();
	return 0;
}
```

Exercise 1:

Problem (1) – update the previous program to do the following:

1. If the letter entered by the user is a lower or upper case letter, the program should retrieve the words that starts by this letter in both upper and lower case.
2. The program should append the found words to the file Lab2P3Words in a new line,

Solution

Problem (2):

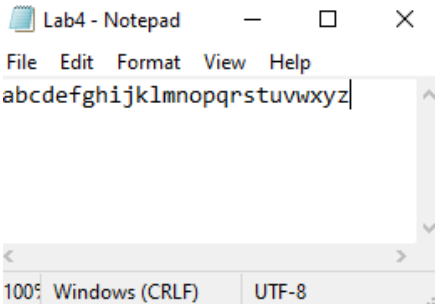
1. Consider you have a file named “Lab4P3Text.txt”, that contains the following text:

   abcdefghijklmnopqrstuvwxyz

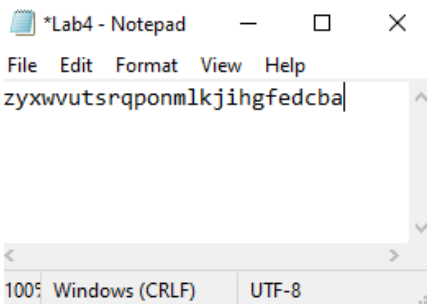
2. Write a C++ program, that reads the text from the file, reverse its order and write it to another file named “Lab4P3RevText.txt”.

Sample of output:

Lab4P3Text.txt



Lab4P3RevText.txt



Solution

Problem (3)

1. You have a file named “Lab4P3Text.txt”, containing a number, e.g. “01236437”. Assume that the number always contains an even number of digits.
2. Write a C++ program that does the following:
    a. Count and print the number of digits in the file. You should count them without using function from the string library.
    b. Prints the sum of the digits in the file.

Hint: Use “stoi(“1”)” to convert from a digit in string format to an integer.

Sample of output:

```
Number of Digits: 8

Sum of digits: 26
```

Solution

Problem (4)

1. You have a file named “Grades.txt”, that contains a series of marks as follows:

01 11 **02** 41 **03** 22 **04** 33 **05** 44 **06** 55 **07** 66 **08** 69 **09** 58 **10** 76

2. Each mark is divided into two components, the module number and the mark of the module.

For example: **01 32**: 01 is the number of the module, and 32 is the mark.

Note: the number of the modules in the file is unknown.

3. Write a C++ program, that calculates and prints on the console the average of the marks and use seekg() function to bypass the module number.

For example: in the above text:

average = ((11+41+22+33+44+55+66+69+58+76)/10).

Solution

```
#include<iostream>
#include<fstream>
#include<string>
using namespace std;
int main() {
	int i = 0, totalMark = 0;
	string mark;
	fstream fin;
	fin.open("GradesFile.txt", ios::in);
	fin.seekg(3, ios::beg);
	while (fin>>mark) {
		totalMark += stoi(mark);
		fin.seekg(3, ios::cur);
		i++;
	}
	fin.close();
	totalMark /= i;
	cout << "Total Mark = " << totalMark;

	return 0;
}
```

Problem (5) - use the previous program as a reference and do the following:

1. You have a file that contains a series of marks. Each mark is of three digits, and has a code represented as a combination of four digits and characters.

For example: **1C1H** 320, the module code is **1C1H**, and the module mark is 320.

Create a file with the name "Grades.txt", and add the following text in it:

1C1H 100 **22PY** 099 **XY12** 022 **31RR** 023 **41RF** 044 **XC33** 050 **W3HG** 060

Note: the number of the modules is unknown.

2. Write a C++ program, that calculate and print on the console the average of the marks. Use seekg() function to bypass the module number.

For example: in the above list:

average = ((100 +099+022+023+044+050+060)/7).

Solution