

Міністерство освіти і науки України

КПІ ім. Ігоря Сікорського

Кафедра ІІІ

ЗВІТ

з виконання лабораторної роботи № 1

з кредитного модуля

“Основи програмування-2. Методології програмування”

Варіант № 22

Виконав:

студент 1-го курсу

гр. ІІ-22 ФІОТ

Підпанюк Віталій Андрійович

Київ 2023

Завдання:

22. Створити текстовий файл. Слова у тексті відділені пробілами, комами, крапками. У кожному рядку даного тексту визначити найдовшу послідовність упорядкованих за зростанням цифр (вважати, що у тексті є такі послідовності). Переставити слова, що містять знайдені послідовності цифр у кінець відповідних рядків. Якщо таких слів у рядку декілька, то переставити ці слова у кінець в порядку їхнього слідування у рядку. Усі змінені рядки вхідного файлу переписати у новий текстовий файл. Замінити відповідні цифри у знайдених словах символами «*». Вивести вміст вхідного і створеного файлів.

Код програми:

main.cpp:

```
#include <iostream>
#include <string>
#include "C++_style.h"
#include "C_style.h"
using namespace std;
int main(int argc, char* argv[]) {
    string argument = argv[1];
    if(argument=="C_style")
    {
        main_capture_c();
    }
    if(argument=="Cpp_style")
    {
        main_capture_cpp();
    }
    cout<<"\nPress eny key to exit\n";
    getchar();
    return 0;
}
```

C++_style.cpp:

```
#include "C++_style.h"
#include "sequence_search.h"
#include <iostream>
#include <string>
#include <istream>
```

```

#include <fstream>
using namespace std;

void main_capture_cpp()
{
    cout<<"You use C++\n";
        file_write_cpp();

    cout<<"*****";
        file_read_cpp();
}

void file_write_cpp()
{
    string iText;
    ofstream oFile("input.txt", ios_base::app);
    if (oFile.is_open())
    {
        do
        {
            getline(cin, iText);
            oFile << iText << '\n';
        } while (!iText.empty());
        oFile.close();
        cout << "Text has been saved in input.txt file." <<
endl;
    }
    else
    {
        cout << "Error: Unable to open file for writing."
<< endl;
    }
}

void file_read_cpp()
{
    string iText;
    string line;
    ifstream iFile("input.txt");

```

```

    if (!iFile.is_open())
        cout << "Error!\nFile does not exist.\n";
    else
    {
        iText = "";
        cout << "Text from input file:" << endl;
        while (!iFile.eof())
        {
            line = "";
            getline(iFile, line);
            iText += line + '\n';
        }
    }
    cout << iText;
    cout << "End of reading\n";
    cout << "Changed text:\n";
    search_cpp(iText);
    iFile.close();
}

```

Cpp_style.h:

```
#pragma once
```

```

void main_capture_cpp();
void file_write_cpp();
void file_read_cpp();

```

C_style.cpp:

```

#include "C_style.h"
#include <iostream>
#include <stdio.h>
#include "sequence_search.h"
using namespace std;

void main_capture_c()
{
    cout<<"You use c style\n";
    //file_write_c();
    file_read_c();
}

```

```

void file_write_c()
{
    cout<<"Your text:\n";
    FILE* filePointer;
    filePointer = fopen("input.txt", "w");
    int size = 0;
    char* dynamicArray = new char [size];
    char element;
    int newQuantity = 4;
    int quantity=newQuantity;
    while (/*cin >> element*/ cin.get(element) && element
!= '^')
    {
        if (size >= quantity)
        {
            quantity += 1;
            char* newDynamicArray = new char[quantity];
            copy(dynamicArray, dynamicArray + size,
newDynamicArray);
            delete[] dynamicArray;
            dynamicArray = newDynamicArray;
        }
        if(element != '^')
        {
            dynamicArray[size] = element;
            size++;
        }
    }
    fprintf(filePointer, "%s", dynamicArray);
    cout<<"Array:\n";
    for (int i = 0; i < size; i++) {
        cout << dynamicArray[i];
    }
    cout << std::endl;
    delete[] dynamicArray;
    fclose(filePointer);
}

```

```

void file_read_c()
{
    FILE* filePointer;
    char fileName[] = "input.txt";
    char character;
    char text[1000];

    filePointer = fopen(fileName, "r");
    if (filePointer == nullptr) {
        std::cout << "File can not be opened!\n";
    }
    else
    {
        cout << "File containing: \n";
        int i = 0;
        while ((character = fgetc(filePointer)) &&
character != EOF) {
            cout << character;
            text[i]=character;
            i++;
        }
        //cout<<"*****\nfile:
" << filePointer;
    }
    fclose(filePointer);
    cout<<"\n\n*****\n";
    cout<<"Changed text:\n";
    search_c(text);
}

```

C_style.h:

```

#pragma once
void main_capture_c();
void file_write_c();
void file_read_c();

```

sequence_search.cpp:

```

#include "sequence_search.h"
#include <iostream>
#include <cstring>

```

```
using namespace std;
```

```
void search_c(char* text) {
    int i = 0;
    char tmp_str[1000] = "";
    char longest_seq[1000] = "";
    char new_string[1000] = "";
    int tmp_index = 0;
    while (text[i] != '\0') {
        if (text[i] != '\n') {
            tmp_str[tmp_index++] = text[i];
        }
        else
        {
            tmp_str[tmp_index] = '\0';
            strcpy(longest_seq, line_check_c(tmp_str));
            if (longest_seq[0] == '\0')
            {
                strcpy(new_string, tmp_str);
            } else
            {
                strcpy(new_string, perform_c(tmp_str,
longest_seq));
            }
            cout << new_string << endl;
            tmp_index = 0;
        }
        i++;
    }
}
```

```
char* line_check_c(char* line) {
    int i = 0, j = 0;
    char* sequence = new char[1000];
    //char sequence[1000] = "";
    for (int k = 0; k < 1000; k++) {
        sequence[k] = '\0';
    }
}
```

```

char alt_sequence[1000] = "0";
while (line[i] != '\0')
{
    if (isdigit(line[i]) && line[i] > alt_sequence[j])
{
        alt_sequence[++j] = line[i];
        i++;
    } else if (isdigit(line[i]) && line[i] <=
alt_sequence[j])
    {
        if (strlen(alt_sequence) > strlen(sequence))
        {
            strcpy(sequence, alt_sequence);
        }
        alt_sequence[0] = '\0';
        i++;
        j = 0;
    }
    else
    {
        if (strlen(alt_sequence) > strlen(sequence))
        {
            strcpy(sequence, alt_sequence);
        }
        alt_sequence[0] = '\0';
        i++;
        j = 0;
    }
}
return sequence;
}

```

```

char* perform_c(char* tmp_str, char* longest_seq) {
    char replacement[1000];
    for (int k = 0; k < strlen(longest_seq); k++) {
        replacement[k] = '*';
    }
    //replacement[strlen(longest_seq)] = '\0';
    char* new_str = new char[1000];
}

```



```

    strcpy(new_str, tmp_str);
    longest_seq[0] = ' ';
    char* ptr = strstr(new_str, longest_seq);
    if (ptr != nullptr) {
        strncpy(ptr, replacement, strlen(longest_seq));
    }
    strcat(new_str, " ");
    strcat(new_str, longest_seq);
    return new_str;
}

```

```

void search_cpp(string text) {
    int i = 0;
    string tmp_str, longest_seq, new_string;
    while (text[i] != '\0')
    {
        if (text[i] != '\n')
        {
            tmp_str += text[i];
        }
        else
        {
            longest_seq = line_check(tmp_str);
            if (longest_seq.empty())
            {
                new_string = tmp_str;
            }
            else
            {
                new_string = perform(tmp_str,
longest_seq);
            }
            cout << new_string << endl;
            tmp_str.clear();
        }
        i++;
    }
}

```

```
}
```

```
string line_check(string line) {  
    int i = 0, j = 0;  
    string sequence, alt_sequence;  
    alt_sequence[0]= '0';  
    sequence = "";  
    alt_sequence = "";  
    while (line[i] != '\0')  
    {  
        if (isdigit(line[i]) && line[i] > alt_sequence[j])  
        {  
            alt_sequence += line[i];  
            j++;  
            i++;  
        }  
        else if (isdigit(line[i]) && line[i] <=  
alt_sequence[j])  
        {  
            if (alt_sequence.length() > sequence.length())  
            {  
                sequence = alt_sequence;  
            }  
            alt_sequence = "";  
            i++;  
            j = 0;  
        }  
        else  
        {  
            if (alt_sequence.length() > sequence.length())  
            {  
                sequence = alt_sequence;  
            }  
            alt_sequence = "";  
            i++;  
            j = 0;  
        }  
    }  
}
```

```
    return sequence;
}
```

```
string perform(string tmp_str, string longest_seq) {
    string replacement(longest_seq.length(), '*');
    size_t i = tmp_str.find(longest_seq);
    tmp_str.replace(i, longest_seq.length(), replacement);
    tmp_str += " ";
    tmp_str += longest_seq;
    return tmp_str;
}
```

sequence_search.h:

```
#pragma once
#include <string>
using namespace std;

void search_c();
void search_cpp(string);
string line_check(string);
string perform(string, string);
void search_c(char*);
char* line_check_c(char*);
char* perform_c(char*, char*);
```

Тестування програми:

```
PS C:\my__folder\study\00P\lab_1\projects_2\project_1\cmake-build-debug> ./project_1.exe Cpp_style
You use C++
abc123

Text has been saved in input.txt file.
*****Text from input file:
mseflkmsl12345kneslfnesl
lmslefm45678n,mfs knkenf
awda123456knfnkanf
kms>ZXux0text

abc123

End of reading
Changed text:
mseflkmsl*****kneslfnesl 12345
lmslefm*****n,mfs knkenf 45678
awda*****knfnkanf 123456
```