МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

КАФЕДРА ТЕОРЕТИЧЕСКОЙ И ПРИКЛАДНОЙ ИНФОРМАТИКИ

ПРАКТИЧЕСКОЕ ЗАДАНИЕ №4

по дисциплине «Информационные технологии и основы программирования»

Факультет: ФПМИ

Группа: ПМИ-33

Студенты: Кауфман Я. Д., Ерощенко А. А.

Преподаватель: Целебровская М.Ю.

НОВОСИБИРСК, 2023

**1) Условие задачи**

Дана непустая последовательность слов (не более 50), в каждом слове не более восьми символов. Слова разделены пробелом, за по-следним словом точка. Найти те слова, которые отличаются от по-следнего слова и удовлетворяют следующему условию:

а) слово симметрично;

б) первая буква слова входит в него еще раз;

в) в слове нет повторяющихся букв;

г) каждая буква входит в слово не менее двух раз.

**2) Анализ задачи**

*Входные данные*: непустая последовательность слов (не более 50), в каждом слове не более восьми символов. Слова разделены пробелом, за по-следним словом точка.

*Выходные данные*: а) симметричные слова;

б) слова, первая буква которых входит в него ещё раз;

в) слова, в которых нет повторяющихся букв;

г) слова, каждая буква которых входит в слово не менее двух раз.

*Метод решения*:

**3)Алгоритм**

**4) Текст программы**

#include <iostream>

#include <io.h>

#include <fcntl.h>

#include <math.h>

#include <vector>

#include <string>

#include <map>

#include <assert.h>

std::vector<std::string> gain\_input(){

std::vector<std::string> out;

out.push\_back(std::string());

char sym;

while(EOF != (sym = fgetc(stdin)) && sym != '.'){

if(sym == ' '){

if(out.rbegin()->length() < 8){

out.push\_back(std::string());

} else {

std::cout << "Word's length should be less than 8!\n";

return {};

}

} else if(sym != '\n'){

\*out.rbegin() += sym;

}

}

if(out.rbegin()->length() >= 8){

std::cout << "Word's length should be less than 8!\n";

return {};

}

for(std::vector<std::string>::iterator it = out.begin(); it != out.end() - 1;){

if(\*it == \*out.rbegin()) it = out.erase(it);

else it ++;

}

return out;

}

char check\_symmetry(std::string& str){

const int half\_len = (str.length() / 2);

for(int i = 0; i < half\_len; i++){

if(\*(str.begin() + i) != \*(str.rbegin() + i)) return 0;

}

return 1;

}

char is\_first\_letter\_doubled(std::string& str){

if(str.length() == 0) return 0;

std::map<char, unsigned int> count;

std::size\_t count\_of\_char;

for(std::string::const\_iterator it = str.begin() + 1; it != str.end(); it++){

count\_of\_char = count.count(\*it);

if(count\_of\_char >= 2 && \*it == \*str.begin()) return 0;

if(count\_of\_char != 0){

count[\*it] = 1;

} else{

count[\*it] ++;

}

}

if(count.count(\*str.begin()) == 1) return 1;

return 0;

}

char not\_double(std::string& str){

std::map<char, int> count;

for(char e: str){

if(count.count(e) ){

return 0;

}else{

count[e] = 1;

}

}

return 1;

}

char is\_letter\_occured\_me2(std::string& str){

if(str.length() == 0) return 0;

std::map<char, unsigned int> count;

for(std::string::const\_iterator it = str.begin(); it != str.end(); it++){

if(count.count(\*it) > 0){

count[\*it] ++;

} else{

count[\*it] = 1;

}

}

for(std::map<char, unsigned int>::const\_iterator it = count.begin(); it != count.end(); it++){

if(it->second < 2) return 0;

}

return 1;

}

int main(){

int v;

std::cout << "Type 1, if you want to find symmetric words. \n"

<< "Type 2, if you want to find words where the first letter occurs one more time. \n"

<< "Type 3, if you want to find words where no letter repeats. \n"

<< "Type 4, if you want to find words where each letter occurs not less than twice. \n";

std::cin >> v;

std::wcout << "Input sequence: \n";

std::vector<std::string> input = gain\_input();

if(input.size() == 0) return EXIT\_FAILURE;

if(v == 1){

for(std::string& word : input){

if(check\_symmetry(word)) std::cout << word <<" ";

}

}else if (v == 2){

for(std::string& word : input){

if(is\_first\_letter\_doubled(word)) std::cout << word <<" ";

}

}else if (v == 3){

for(std::string& word : input){

if(not\_double(word)) std::cout << word <<" ";

}

}else if (v == 4){

for(std::string& word : input){

if(is\_letter\_occured\_me2(word)) std::cout << word <<" ";

}

}

return 0;

}

**5) Набор тестов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Входные данные  v | Входные данные | Выходные данные | Примечание |
| 1 | 1 | assa boob toto. | assa boob |  |
| 2 | 2 | hello mom me. | mom |  |
| 3 | 3 | assa mom me said. | me said |  |
| 4 | 4 | asssaa. | asssaa |  |
| 5 | 2 | strengths and me. | Word's length should be less than 8! |  |

**6) Результаты отладки**

Задача решена правильно, что подтверждается результатами тестов.