Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра электронных вычислительных машин

ОТЧЕТ

О ЛАБОРАТОРНОЙ РАБОТЕ № 2

Дружественные функции. Перегрузка операций

по дисциплине «Программирование на языках высокого уровня»

Выполнил ст. гр. 450503 Д.С. Абрамчук

Проверил асс. каф. ЭВМ И.Г. Скиба

Минск 2025

# **1 ПОСТАНОВКА ЗАДАЧИ**

Реализовать класс String для работы со строками символов. Перегрузить операторы =, +=,  ==. Предоставить конструктор копирования. Определить friend функции для операций ввода-вывода в поток.

# **2 ЛИСТИНГ КОДА**

Файл main.cpp

#include "function.h"  
  
int main() {  
 ShowMenu();  
 return 0;  
}

Файл function.cpp

#include "function.h"  
  
  
String::String() : text(new char[1]) {  
 text[0] = '\0';  
}  
  
  
String::String(const char \*str) {  
 if (str) {  
 text = new char[strlen(str) + 1];  
 strcpy(text, str);  
 } else {  
 text = new char[1];  
 text[0] = '\0';  
 }  
}  
  
String::String(const String &other) {  
 text = new char[strlen(other.text) + 1];  
 strcpy(text, other.text);  
}  
  
String::~String() {  
 delete[] text;  
}  
  
String &String::operator=(const String &other) {  
 if (this != &other) {  
 delete[] text;  
 text = new char[strlen(other.text) + 1];  
 strcpy(text, other.text);  
 }  
 return \*this;  
}  
  
String &String::operator+=(const String &other) {  
 auto temp = new char[strlen(text) + strlen(other.text) + 1]; // auto вместо char\*  
 std::strcpy(temp, text);  
 std::strcat(temp, other.text);  
 delete[] text;  
 text = temp;  
 return \*this;  
}  
  
  
bool String::operator==(const String &other) const {  
 return strcmp(text, other.text) == 0;  
}  
  
void ShowMenu() {  
 String s1;  
 String s2;  
 int choice;  
  
 do {  
 std::cout << "\n MENU \n";  
 std::cout << "1. Input first string\n";  
 std::cout << "2. Display first string\n";  
 std::cout << "3. Display second string\n";  
 std::cout << "4. Assign first string to second\n";  
 std::cout << "5. Add second string to first\n";  
 std::cout << "6. Check if strings are equal\n";  
 std::cout << "0. Exit\n";  
 std::cout << "Choose point: ";  
 std::cin >> choice;  
  
 switch (choice) {  
 case 1:  
 std::cout << "Enter first string: ";  
 std::cin >> s1;  
 break;  
 case 2:  
 std::cout << "First string: " << s1 << std::endl;  
 break;  
 case 3:  
 std::cout << "Second string: " << s2 << std::endl;  
 break;  
 case 4:  
 s2 = s1;  
 std::cout << "Second string contains the first string.\n";  
 break;  
 case 5:  
 s1 += s2;  
 std::cout << "First string after adding second: " << s1 << std::endl;  
 break;  
 case 6:  
 if (s1 == s2)  
 std::cout << "Strings are equal.\n";  
 else  
 std::cout << "Strings are different.\n";  
 break;  
 case 0:  
 std::cout << "Exit program.\n";  
 break;  
 default:  
 std::cout << "ERROR. Try again.\n";  
 }  
 } while (choice != 0);  
}

Файл function.h

#ifndef LAB\_\_\_2\_\_FUNCTION\_H  
#define LAB\_\_\_2\_\_FUNCTION\_H  
  
#include <iostream>  
#include <cstring>  
  
class String {  
 char \*text;  
  
public:  
 String();  
  
 explicit String(const char \*str);  
  
 String(const String &other);  
  
 ~String();  
  
 String &operator=(const String &other);  
  
 String &operator+=(const String &other);  
  
 bool operator==(const String &other) const;  
  
 friend std::ostream &operator<<(std::ostream &out, const String &str) {  
 out << str.text;  
 return out;  
 }  
  
 friend std::istream &operator>>(std::istream &in, String &str) {  
 std::string temp;  
 in >> temp; // безопасный ввод, без ограничения длины  
 delete[] str.text;  
 str.text = new char[temp.size() + 1];  
 std::memcpy(str.text, temp.c\_str(), temp.size() + 1);  
 return in;  
 }  
};  
  
void ShowMenu();  
  
#endif

**3 РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ**

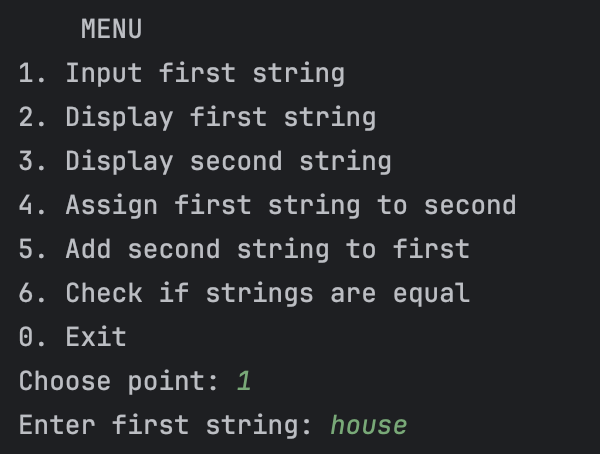


Рисунок 4.1 – ввод первой строки

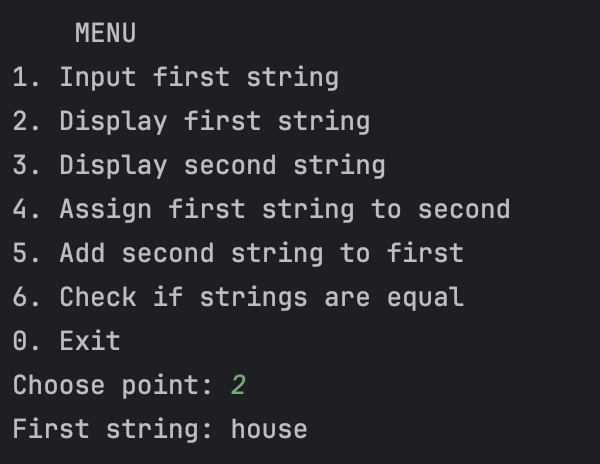


Рисунок 4.2 – демонстрация первой строки

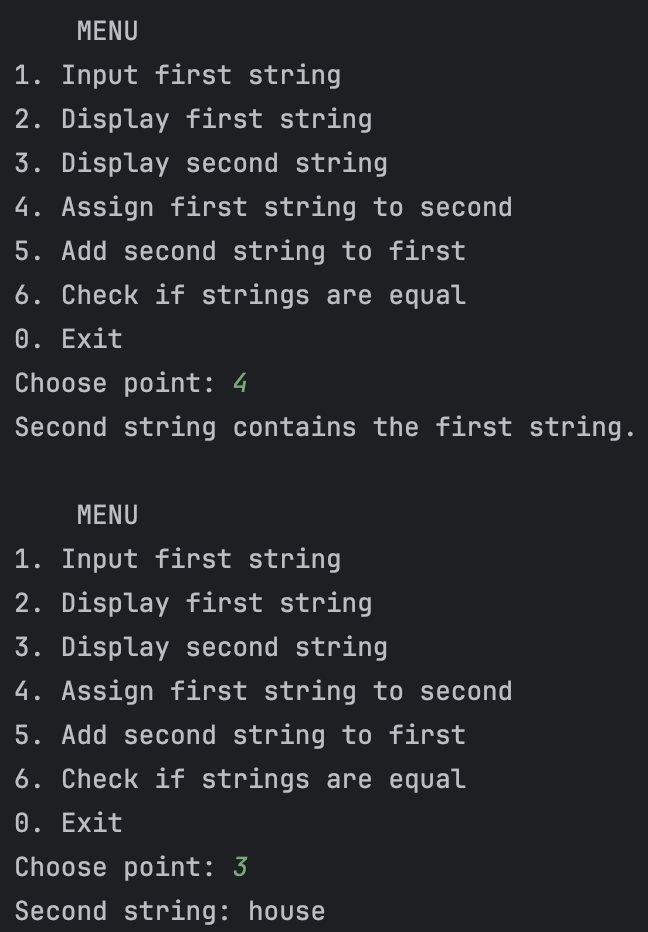


Рисунок 4.3 – демонстрация второй строки, после копирования

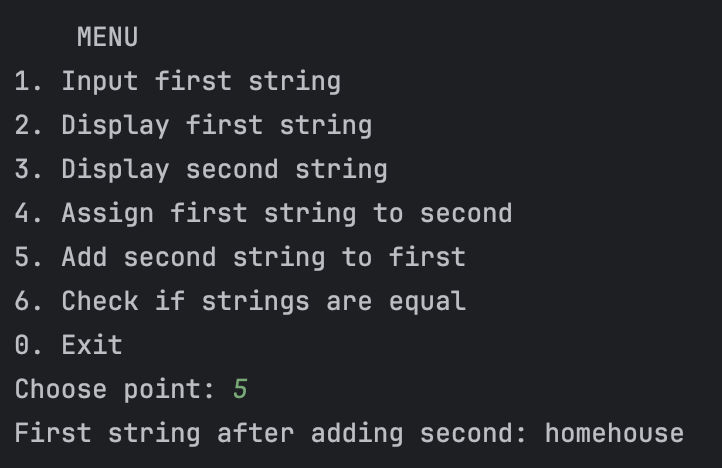


Рисунок 4.4 – объединение строк

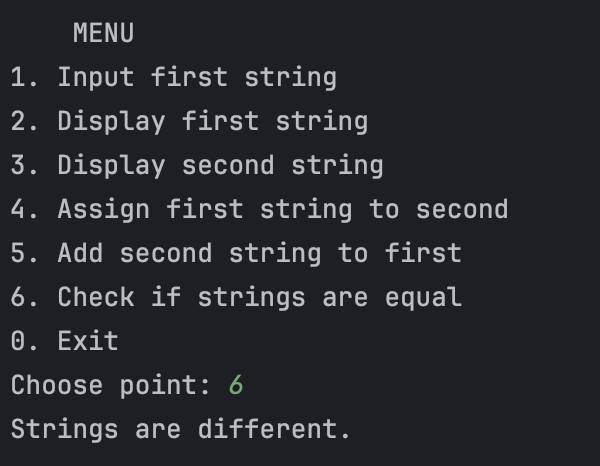


Рисунок 4.5 – проверка равны ли строки