Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра **«Информационные технологии и автоматизированные системы»**

направление подготовки: 09.03.04 - «Программная инженерия»

Лабораторная работа №6

По теме **«Строки»**

**Вариант № 21**

Выполнял:

студент группы РИС-24-1б

Морозова Н.С.

Проверял:

Доцент кафедры ИТАС

Полякова О.А.

Постановка задачи: задана строка, состоящая из символов. Символы объединяются в слова. Слова друг от друга отделяются одним или несколькими пробелами (символом подчёркивания ‘\_’). В конце текста ставится точка. Текст содержит не более 255 символов. Выполнить ввод строки, используя функцию getline() и обработку строки в соответствии со своим вариантом.

С клавиатуры вводятся:

1. Предложение, слова в котором разделены символом подчёркивания (‘\_’);
2. Маска для выбора из предложения нужных слов (содержит буквы и символ заполнитель ‘\*’, который заменяет любое сочетание букв, в том числе пустое).

Необходимо выбрать из предложения все слова, соответствующие маске.

Допустимые символы - русские буквы, символ разделитель ‘\_’.

Код

#include <iostream>

#include <clocale>

#include <string>

#include <vector>

#include <Windows.h>

using namespace std;

// Функция для проверки, соответствует ли слово маске

bool isMatch(const string& word, const string& mask)

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

size\_t wordPos = 0; // Позиция в слове

size\_t maskPos = 0; // Позиция в маске

size\_t starIdx = string::npos; // Хранит индекс последней звезды

size\_t matchIdx = 0; // Хранит индекс совпадения в слове

while (wordPos < word.size())

{

// Если символы совпадают или маска содержит '?'

if (maskPos < mask.size() && (mask[maskPos] == '?' || mask[maskPos] == word[wordPos]))

{

wordPos++;

maskPos++;

}

// Если встречаем '\*', запоминаем положение

else if (maskPos < mask.size() && mask[maskPos] == '\*')

{

starIdx = maskPos;

matchIdx = wordPos;

maskPos++;

}

// Если не совпало, возвращаемся к последней '\*'

else if (starIdx != string::npos)

{

maskPos = starIdx + 1;

wordPos = ++matchIdx; // Пробуем следующее символ в слове

}

else

return false; // Если символ не совпадает и не осталось '\*'

}

while (maskPos < mask.size() && mask[maskPos] == '\*') // Пропускаем оставшиеся '\*' в маске

maskPos++;

return maskPos == mask.size();

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(0, "");

string text;

string mask;

cout << "Введите продложение (на английском): ";

getline(cin, text);

cout << "Введите маску (используйте '\*' и '?' ): ";

getline(cin, mask);

// Удаляем точку в конце строки, если она есть

if (!text.empty() && text.back() == '.')

text.pop\_back();

vector<string> words; // Список слов

size\_t start = 0; // Начало текущего слова

size\_t end = text.find('\_'); // Позиция символа '\_'

// Разделяем строку на слова по символу '\_'

while (end != string::npos)

{

words.push\_back(text.substr(start, end - start));

start = end + 1;

end = text.find('\_', start);

}

// Добавляем последнее слово

words.push\_back(text.substr(start));

cout << "Слова, подходящие маске: ";

for (const string& w : words)

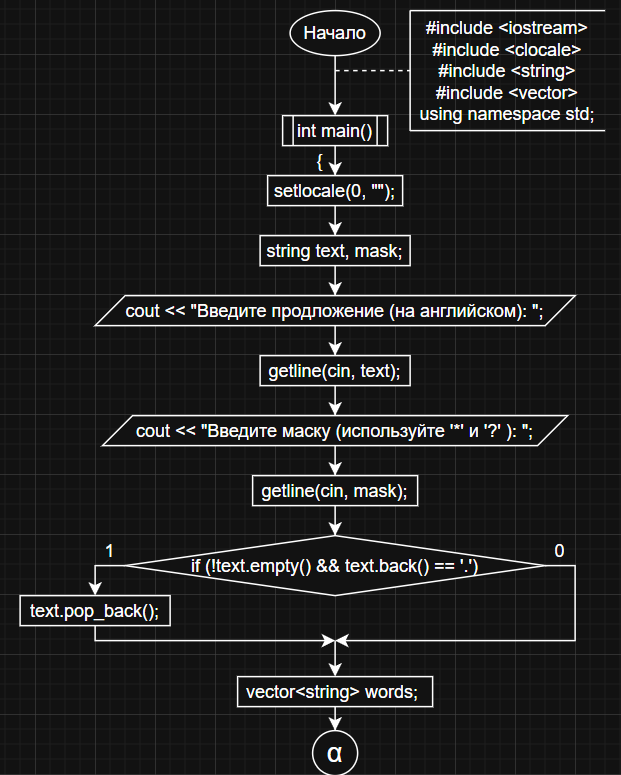
if (isMatch(w, mask))

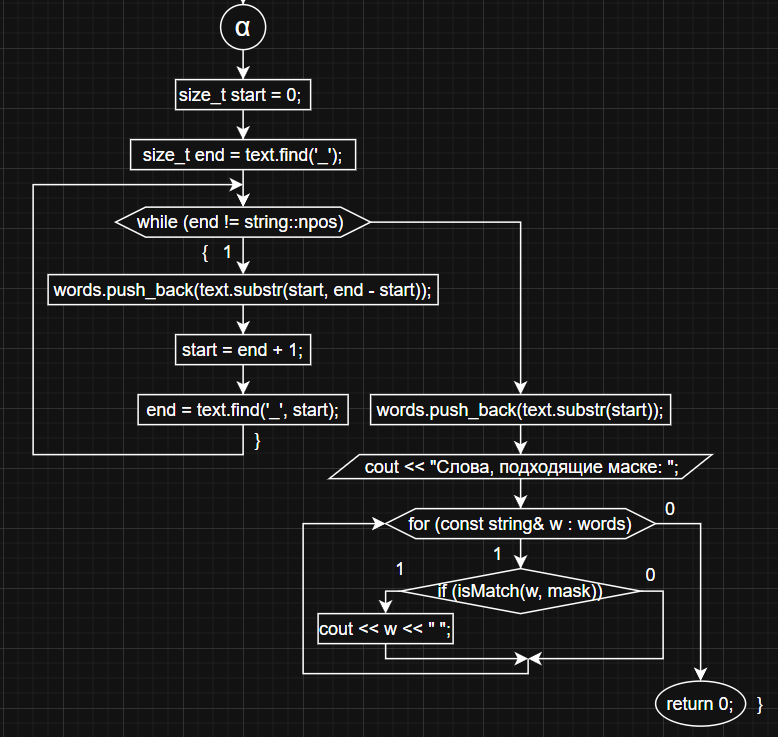
cout << w << " ";

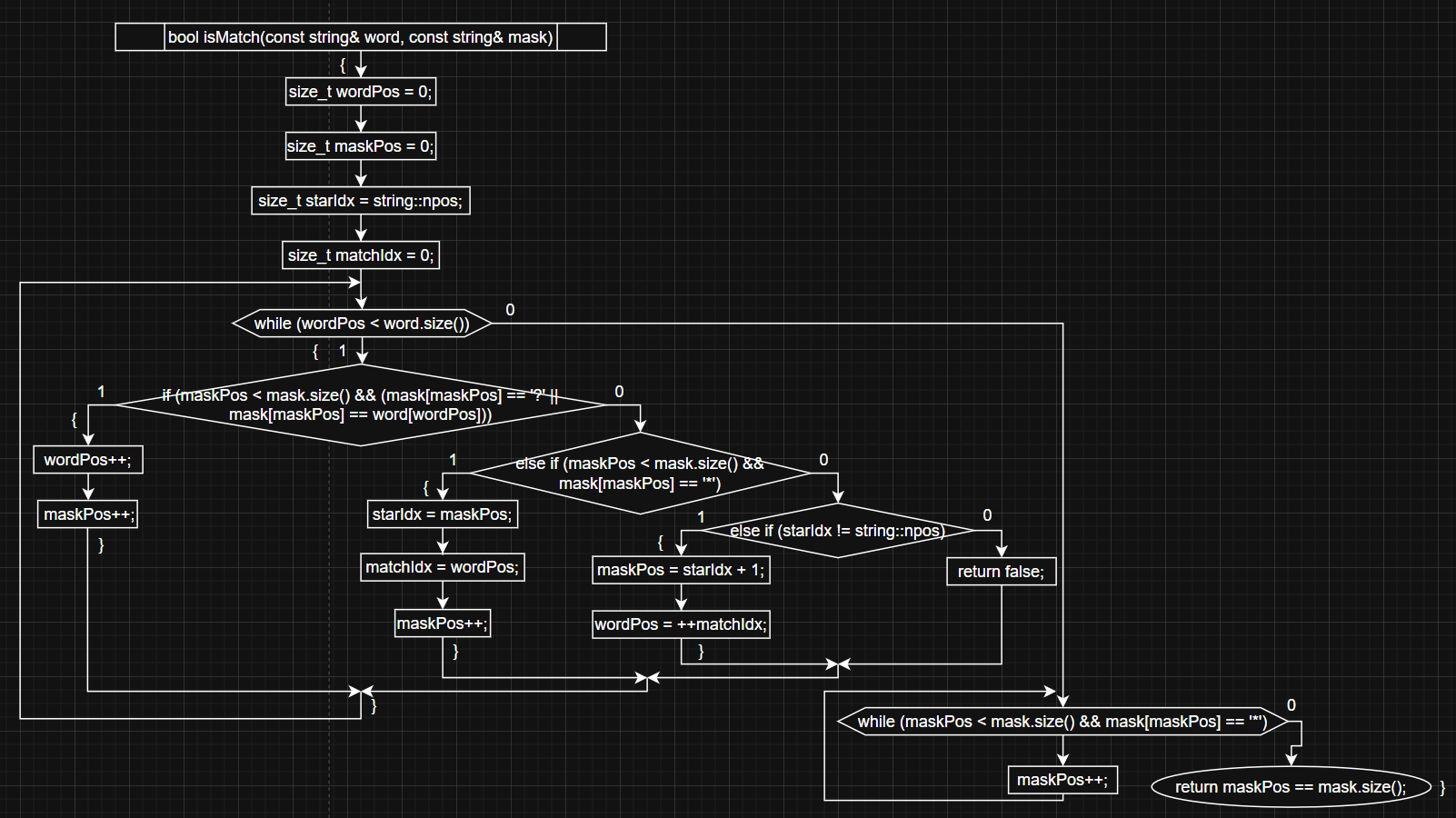
return 0;

}

Блок-схема







Примеры работы программы:

