Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

(ПНИПУ)

ОТЧЁТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

по теме:

СТРОКИ

Выполнила: студентка группы РИС-22-1б

Черкасова А.А\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил: доцент кафедры ИТАС

Полякова О.А\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_/\_\_\_\_\_\_\_/\_\_\_\_\_г.

Пермь 2023

**Содержание**

[**Введение** 3](#_Toc127379672)

[**Анализ задачи** 3](#_Toc127379673)

[**Блок – схема** 5](#_Toc127379674)

[**Приложение А** 8](#_Toc127379675)

[**Приложение Б** 10](#_Toc127379676)

# **Введение**

**Постановка задачи:** Задана строка, состоящая из символов. Символы объединяются в слова. Слова друг от друга отделяются одним или несколькими пробелами. В конце текста ставится точка.

Все слова строки, которые начинаются с буквы, отсортировать в алфавитном порядке.

**Цель:** Изучение символьных и строковых переменных и способов их обработки в языке Си.

Для достижения поставленной цели, необходимо решить следующие **задачи**:

* Провести анализ задачи
* Реализовать задачу на языке С++
* Составить блок-схему

# **Анализ задачи**

1. В функции *main* пользователь задаёт строку. Регистром букв в словах можно пренебрегать. Важно, чтобы строка состояла из русских символов.

2. Введённая строка помещается в качестве параметра в функцию *ArrayWords.* В этой функции с помощью цикла *for* программа проходится по каждому символу строки, помещая эти символы в переменную типа *string,* до тех пор, пока не встретит пробел. Как только встречается пробел, это означает, что слово закончилось. После этого, выделенное из строки слово помещается в переменную типа *vector (в конец).*

3. После того, как строка закончилась и переменная типа *vector* заполнена, происходит передача этой переменной в качестве параметра в функцию *SORT.* Здесь происходит сортировка методом обмена, но сравниваются здесь коды символов в соответствии с таблицей ASCII.

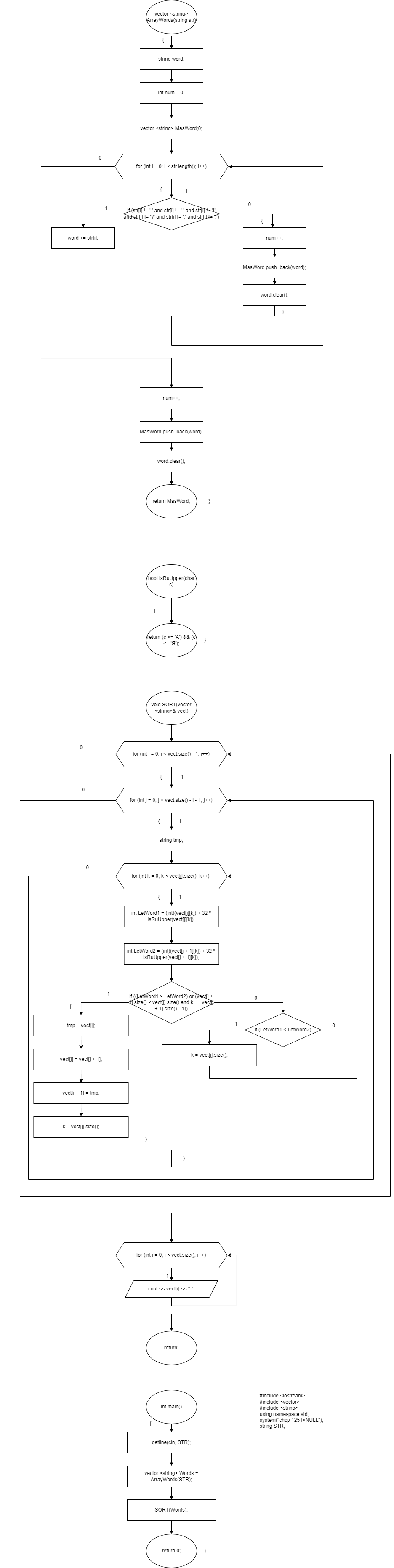
5. Если код первого символа первого слова больше, чем код первого символа второго слова, то слова меняются местами. И происходит принудительный выход из цикла, перебирающего символы слова. Если первые символы одинаковы, то сравниваются вторые символы каждого слова и т.д.

6. Регистр не важен, т.к. он учитывается в выражении:

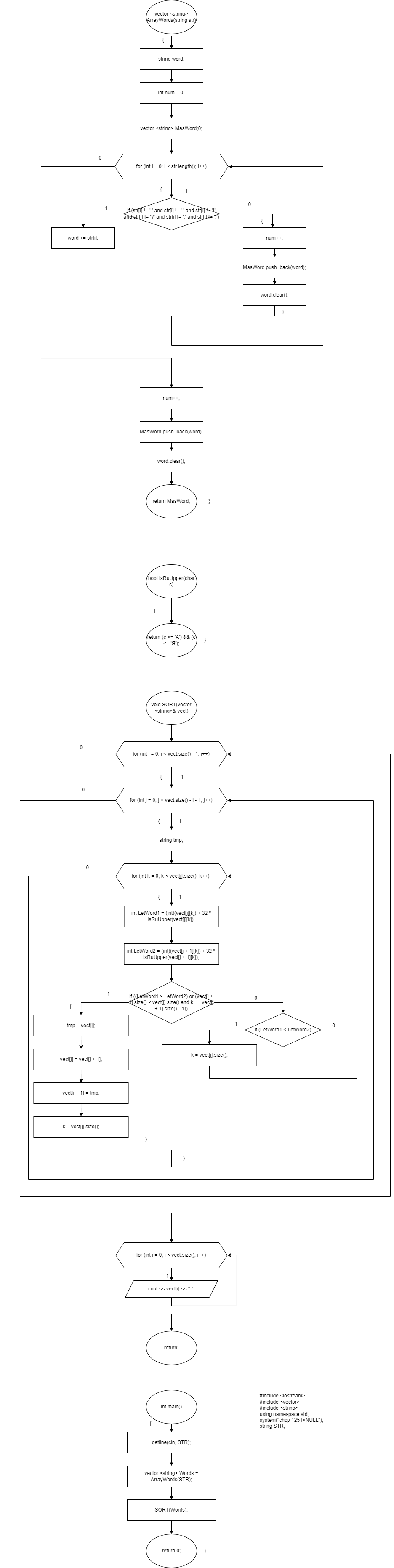
*(int)(vect[j][k]) + 32 \* IsRuUpper(vect[j][k]),* т.е. чтобы узнать код текущего символа необходимо этот символ типа *char*, привести к типу *int.* В данном выражении сравниваются условно строчные буквы, поэтому, если буквы будет прописные, то произойдёт вызов функции *IsRuUpper,* возвращающей значения 1 или 0, и к коду прописной буквы прибавится 32. Таким образом происходит преобразование прописных букв к строчным для сравнения.

# **Блок – схема**

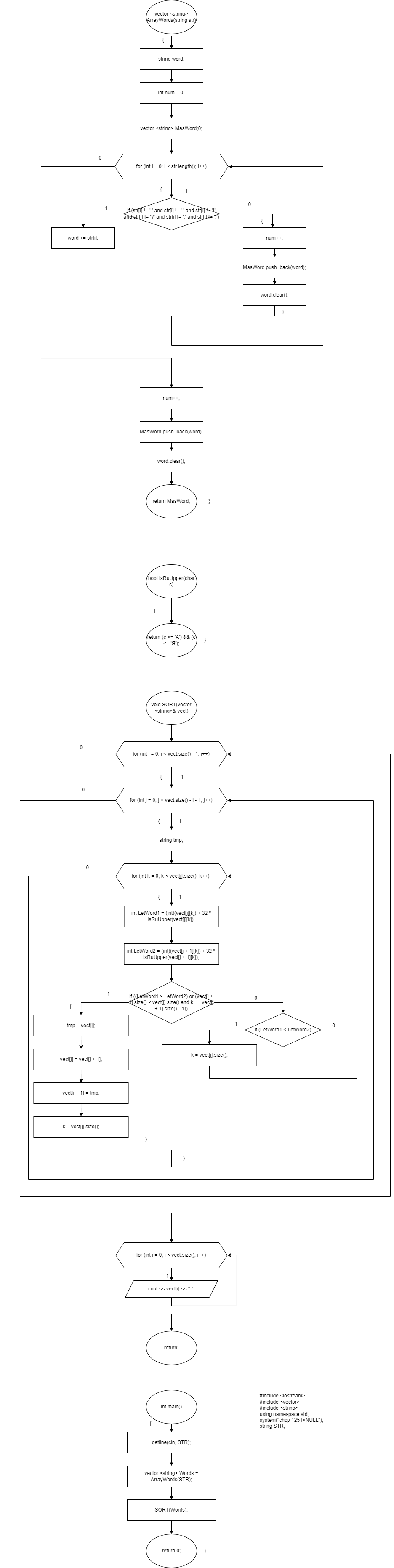
A) Функция *ArrayWords:*

**

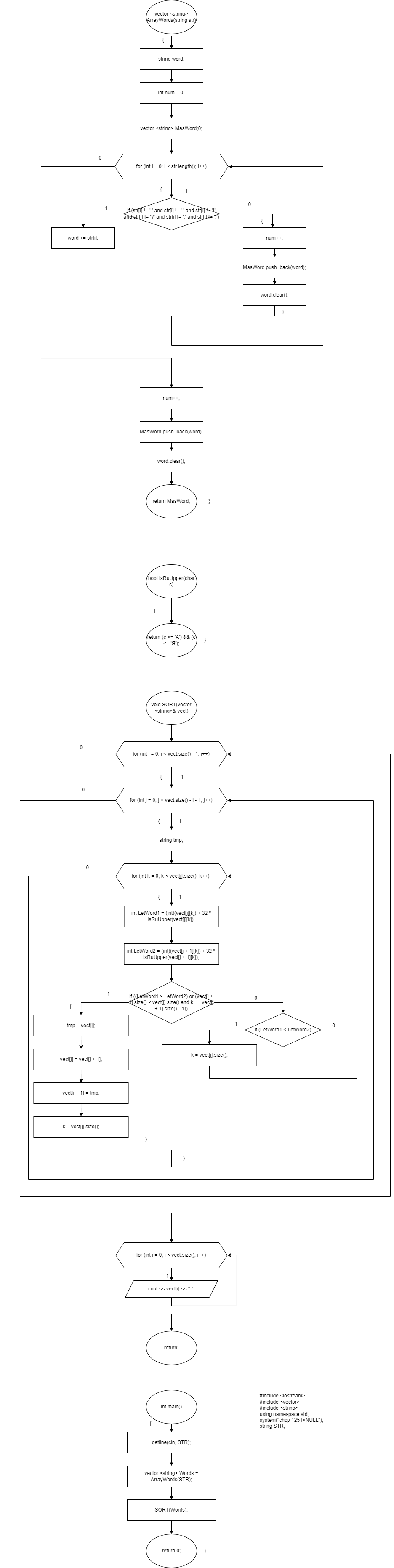
B) Функция *IsRuUpper:*

**

C) Функция *main:*

**

D) Функция *SORT:*

**

# **Приложение А**

**Листинг программы**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

vector <string> ArrayWords(string str) // создание массива слов

{

string word; // "временное" слово

int num = 0; // длина массива

vector <string> MasWord;

for (int i = 0; i < str.length(); i++) // проход по всей длине строки

{

if (str[i] != ' ' and str[i] != '.' and str[i] != '!' and str[i] != '?' and str[i] != ':' and str[i] != ';')// "вычленение" слова из строки без учета знаков

word += str[i]; // составление слова побуквенно

else

{

num++;

MasWord.push\_back(word);

word.clear();

}

}

num++;

MasWord.push\_back(word); // добавление слова в конец массива

word.clear();

return MasWord;

}

bool IsRuUpper(char c) // проверка русской буквы на регистр

{

return (c >= 'А') && (c <= 'Я');

}

void SORT(vector <string>& vect) // сортировка пузырьком

{

for (int i = 0; i < vect.size() - 1; i++)

{

for (int j = 0; j < vect.size() - i - 1; j++) // n - i = длина неотсортированной части

{

string tmp;

for (int k = 0; k < vect[j].size(); k++) // цикл, отвечающий за букву в слове

{

int LetWord1 = (int)(vect[j][k]) + 32 \* IsRuUpper(vect[j][k]); // преобразование к одному регистру букв слов

int LetWord2 = (int)(vect[j + 1][k]) + 32 \* IsRuUpper(vect[j + 1][k]);

if ((LetWord1 > LetWord2) or (vect[j + 1].size() < vect[j].size() and k == vect[j + 1].size() - 1)) // проверка на длины, есличасти слова одинаковая

{

tmp = vect[j];

vect[j] = vect[j + 1];

vect[j + 1] = tmp;

k = vect[j].size(); // принудительный выход

}

if (LetWord1 < LetWord2) // проверка на длины, есличасти слова одинаковая

{

k = vect[j].size(); // принудительный выход

}

}

}

}

cout << "--------------------------------- SORT\_VECTOR --------------------------------------" << endl;

for (int i = 0; i < vect.size(); i++) cout << vect[i] << " "; // вывод массива

cout << endl;

}

int main()

{

cout << "--------------------------------- STRING --------------------------------------" << endl;

system("chcp 1251>NULL");

string STR;

cout << "Enter STRING: ";

getline(cin, STR);

cout << endl;

vector <string> Words = ArrayWords(STR);

SORT(Words);

}

# 

# **Приложение Б**

**Результаты выполнения программы**

