Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЁТ

к лабораторной работе №8

на тему

РАБОТА СО СТРОКАМИ

ВАРИАНТ 6

Проверил Выполнил

асс. каф. ЭВМ студент гр. 318302 П.С. Жуковец В.С.Бахур

МИНСК 2023

**1** **ЗАДАНИЕ №1**

**1.1 Условие**

Отсортировать методом вставок массив строк по убыванию длины максимального слова в каждой строки

**1.2 Алгоритм решения**

Алгоритм предоставлен в «Приложении A»; Рисунок 1.1

**1.3 Листинг программы**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

// Функция для нахождения длины самого длинного слова в строке

int longestWordLength(char\* str) {

int maxLen = 0, currentLen = 0;

while (\*str) {

if (\*str == ' ' || \*str == '\n' || \*str == '\t') {

if (currentLen > maxLen) {

maxLen = currentLen;

}

currentLen = 0;

}

else {

currentLen++;

}

str++;

}

if (currentLen > maxLen) {

maxLen = currentLen;

}

return maxLen;

}

// Функция сортировки массива строк методом вставки

void insertionSort(char\*\* array, int n) {

for (int i = 1; i < n; i++) {

char\* temp = array[i];

int j = i - 1;

while (j >= 0 && longestWordLength(array[j]) < longestWordLength(temp)) {

array[j + 1] = array[j];

j--;

}

array[j + 1] = temp;}

}

int main(int argc, char\*\* argv) {

system("chcp 1251 > nul");

if (argc > 1) {

insertionSort(argv + 1, argc - 1);

printf("Отсортированный массив строк:\n");

for (int i = 1; i < argc; i++) {

printf("%s\n", argv[i]);

}

}else {printf("Строки не были предоставлены.\n");

}

return 0;}

**Приложение А**

Блок-схема алгоритма 1

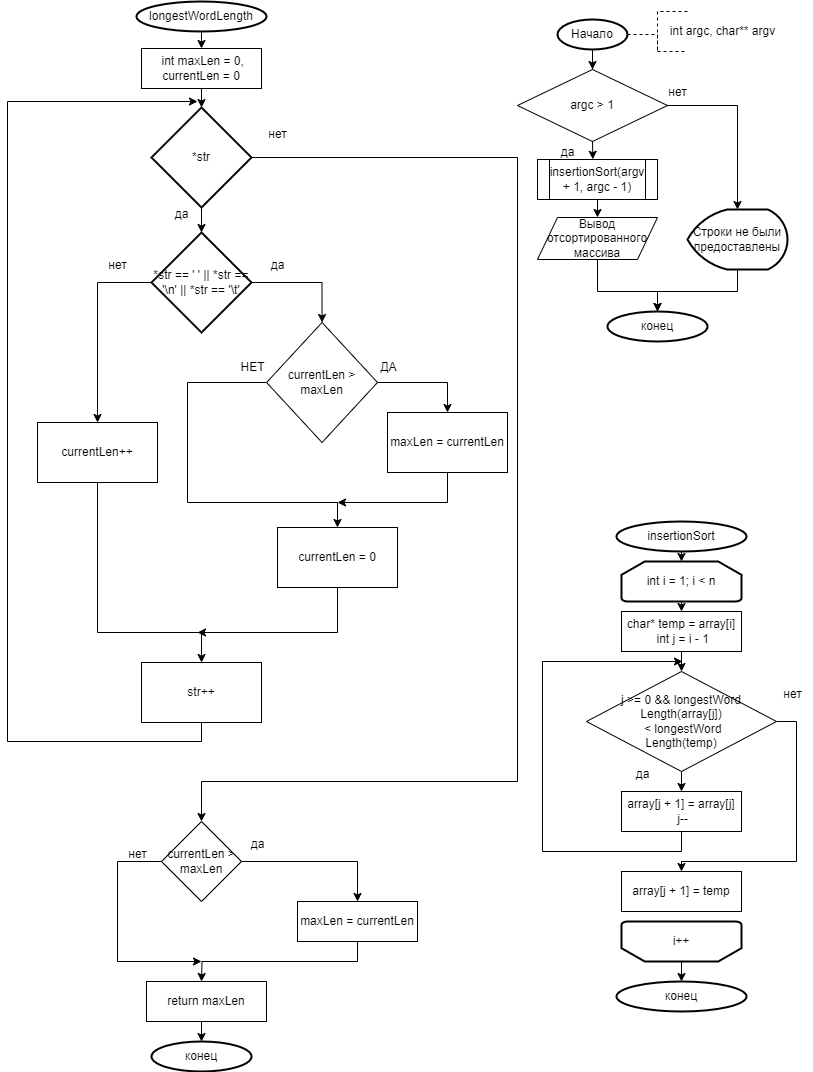


Рисунок 1.1 – Блок-схема №1