Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

Восточно-Сибирский государственный университет технологий и управления

(ФГБОУ ВСГУТУ)

Электротехнический факультет

Кафедра «Системы Информатики»

Дисциплина

«Основы программирования»

Отчет по лабораторной работе №8

На тему:

Динамическое распределение памяти

Вариант3

|  |  |
| --- | --- |
| Выполнил: | студент гр. Б661-2 |
|  | Коковихин А.В. |
| Проверил: | преподаватель  Жамбалов Э.Б. |

Улан-Удэ

2021

# 1 Словесная постановка задачи

Напишите программу, решающую задачу, соответствующую вашему индивидуальному заданию. Для решения задачи необходимо использовать указатели.

Во всех программах необходимо использовать динамические массивы. Размер массива должен вводить пользователь. Для работы с массивами используйте адресную арифметику.

Все изменения массивов и строк в функциях должны сохраняться. Если сказано, что значение сохраняется в переменной-аргументе, то это значит, что все изменения с этой переменной должны отразиться на оригинальной переменной, переданной в функцию, а сама переменная-аргумент является указателем. Все результаты работы программ необходимо выводить на экран.

1. Написать функцию, которая принимает строку и переменную-указатель. Делит строку на две строки. Первая – первая половина исходной строки, и она запоминается в переменной-аргументе. Вторая – вторая половина, и она возвращается функцией в качестве результата.

# 2 Решение задачи 1

## 2.1 Формальная постановка задачи

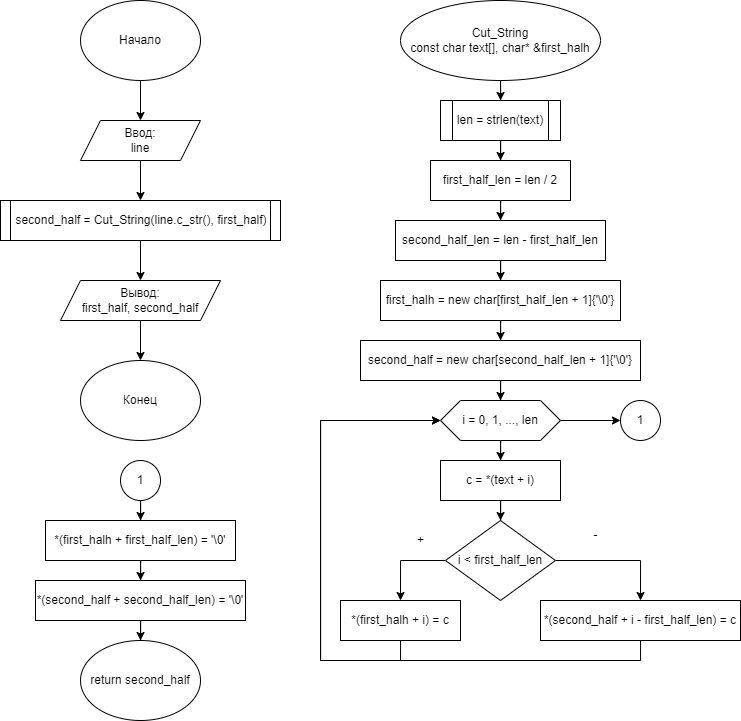
Входные данные:

string line – Вводимая пользователем строка

Выходные данные:

char\* first\_half, second\_half – половины строки.

## 2.2 Проектирование



## 2.3 Реализация

char\* Cut\_String(const char text[], char\* &first\_halh)

{

int len = strlen(text);

int first\_half\_len = len / 2;

int second\_half\_len = len - first\_half\_len;

first\_halh = new char[first\_half\_len + 1]{'\0'};

char\* second\_half = new char[second\_half\_len + 1]{'\0'};

for (int i = 0; i < len; i++)

{

char c = \*(text + i);

if (i < first\_half\_len)

\*(first\_halh + i) = c;

else

\*(second\_half + i - first\_half\_len) = c;

}

\*(first\_halh + first\_half\_len) = '\0';

\*(second\_half + second\_half\_len) = '\0';

return second\_half;

}

string line;

cout << "Enter some text:";

getline(cin, line);

char\* first\_half;

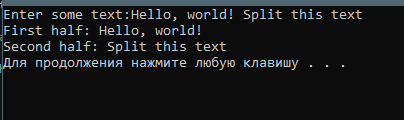
char\* second\_half;

second\_half = Cut\_String(line.c\_str(), first\_half);

cout << "First half: " << first\_half << "\nSecond half: " << second\_half;

delete[] second\_half, first\_half;

## 2.4 Тестирование



**Заключение**

Динамическое выделение памяти позволяет выделять необходимое количество информации на ходу. Отработать эти знания на строковых массивах – вполне наглядно.