Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №10**

Дисциплина: основы алгоритмизации и программирования

Тема: “ Динамические массивы” в C++

Вариант 24

Выполнил работу

студент группы РИС-20-1б

Коваленко Н. А.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2020

**Цель задачи**

Цель – написать программу для создания, редактирования и вывода динамического массива.

Для достижения поставленной цели необходимо решить следующие задачи:

* Проанализировать инициализацию динамического массива
* Продумать алгоритм, который будет отвечать за редактирование
* Проанализировать, как динамическая память влияет на оперативную память, а также понятия куча и стек.
* Проанализировать очистку оперативной памяти

Постановка задачи

Сформировать массив строк. Удалить из него k первых строк.

Анализ задачи

1. Определить какие операции должны быть выполнены по заданию:

* Создание динамического массива заданной длины
* Реализация проверки входных данных
* Заполнение массива
* Редактирование массива, а именно удаление k первых элементов
* Копирование отредактированного массива в новый массив меньшей длины
* Отчистка памяти, выделенной под изначальный массив
* Вывод нового массива

1. Для решения задачи используются переменные:

Несколько целочисленных переменных для управления циклами, и контроля размеров массивов:

int size = -2, k = -5;

while (size < 1)

{

cout << "Введи размерность массива: "; cin >> size;

}

Два указателя на динамические массивы строк

string\* a1 = new string[size];

string\* a2 = new string[size-k];

Два динамических массива: первый изначальный массив с исходными данными, второй с меньшей длиной, в который первый массив будет скопирован:

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

{

cout << "Введи " << i + 1 << " строку:" << "\t"; cin >> a1[i];

}

for (int i = 0; i < size - k; i++)

1. Ввод данных осуществляется посредством функции

cin >> a1[i];

Вывод данных осуществляется посредством функции

cout << i + 1 << ") " << a2[i] << endl;

Кроме функций для ввода и вывода данных используются циклы, которые передвигают указатель на следующую область памяти (следующий элемент массива)

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

{

cout << "Введи " << i + 1 << " строку:" << "\t"; cin >> a1[i];

}

1. Для решения задачи будут использованы циклы. Например, цикл, который редактирует массив, удаляя k первых элементов:

for (int j = 1; j <= k; j++)

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

a1[i] = a1[i + 1];

1. Для защиты программы от случайных ошибок были реализованы проверки входных данных:

Проверка размерности массива (размер больше 0):

while (size < 1)

{

cout << "Введи размерность массива: "; cin >> size;

}

Проверка количества удаляемых строк:

while (k < 0 || k > size)

{

cout << "Введи, сколько строк нужно удалить: "; cin >> k;

}

Код

#include <iostream>

#include <string>

#include <Windows.h>

using namespace std;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int size = -2, k = -5;

while (size < 1)

{

cout << "Введи размерность массива: "; cin >> size;

}

string\* a1 = new string[size];

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

{

cout << "Введи " << i + 1 << " строку:" << "\t"; cin >> a1[i];

}

while (k < 1 || k > size)

{

cout << "Введи, сколько строк нужно удалить: "; cin >> k;

}

for (int j = 1; j <= k; j++)

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

a1[i] = a1[i + 1];

string\* a2 = new string[size-k];

for (int i = 0; i < size - k; i++)

a2[i] = a1[i];

delete[] a1;

cout << "Массив с изменениями: " << endl;

for (int i = 0; i < size - k; i++)

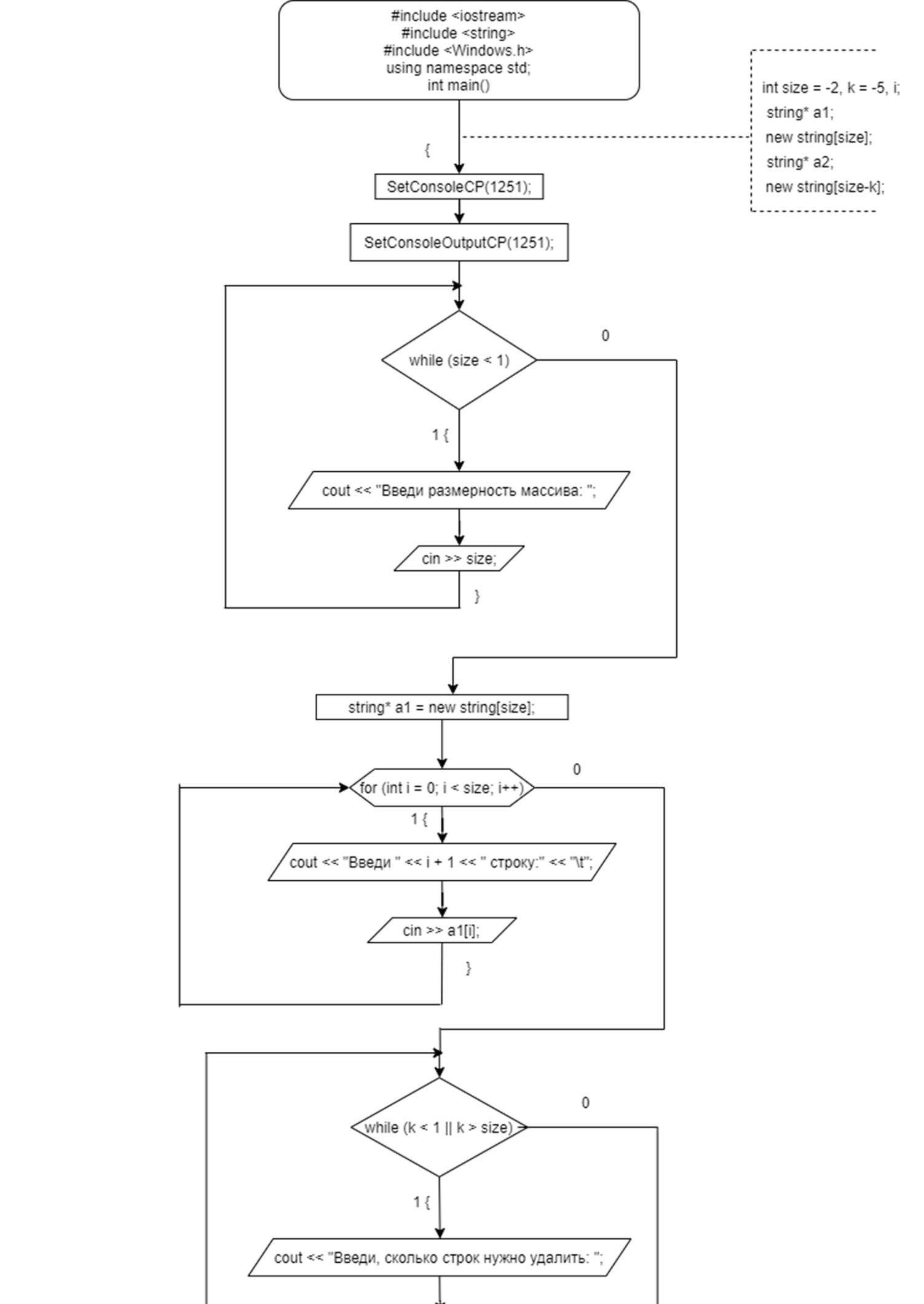
{

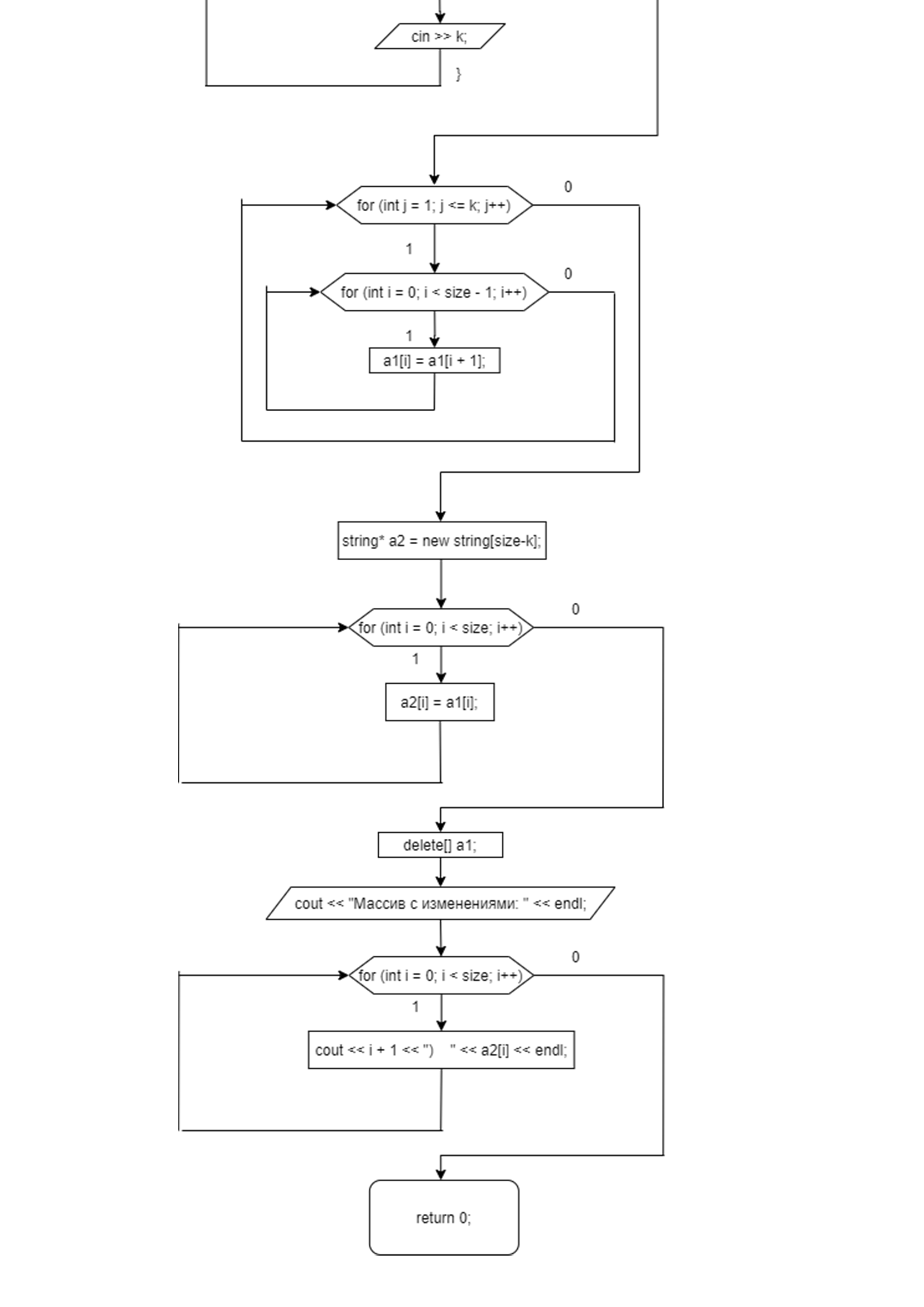
cout << i + 1 << ") " << a2[i] << endl;

}

}

Блок-схема:





Работа кода

