Министерство высшего образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет» (ПНИПУ)**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

ОТЧЁТ

по лабораторной работе №10

Тема: «Динамические массивы»

Выполнил

Студент группы РИС-22-2б

Будин Д.В.

Проверил доц. Кафедры ИТАС

Полякова Ольга Андреевна

Пермь 2023

Постановка задачи

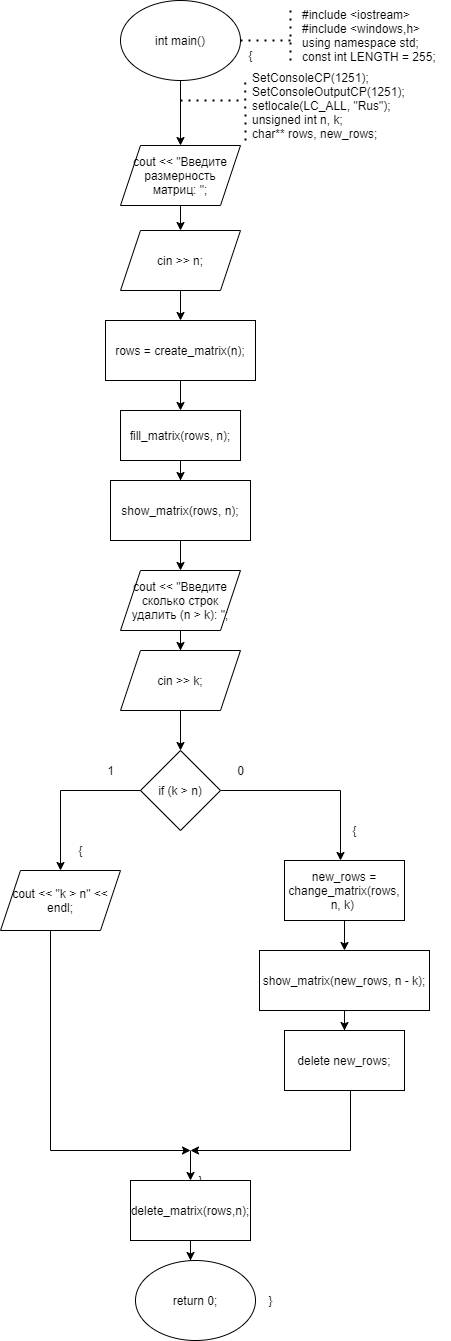
Написать программу, в которой создаются динамические массивы и выполнить их обработку в соответствии со своим вариантом.

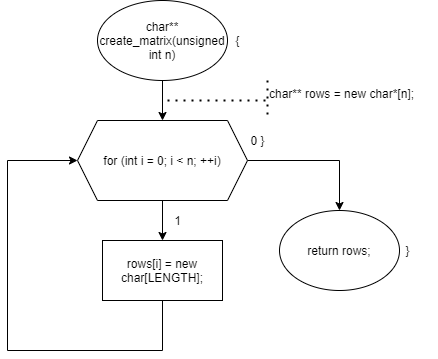
24 ВАРИАНТ.

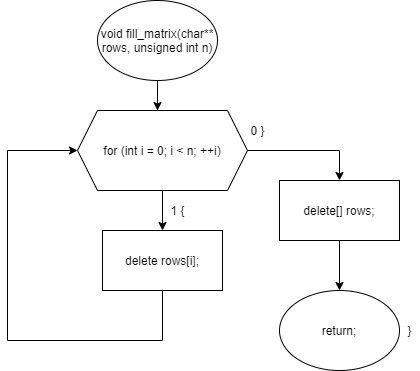
Сформировать массив из строк, удалить из него K первых строк.

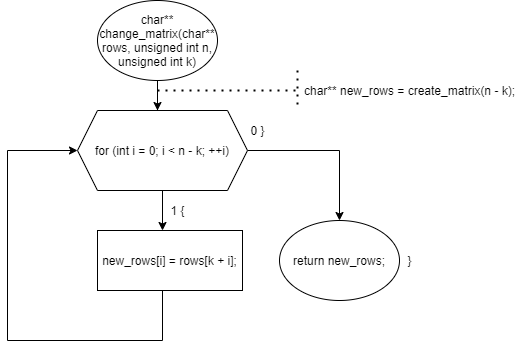
Анализ задачи

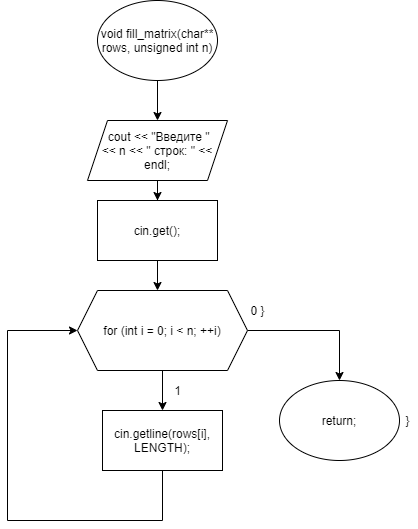
1. Устанавливаем кодировку консоли для правильного вывода строк, также ставим локаль на русскую;
2. Выделяем память под переменные int n, k, где n – кол-во строк, k – кол-во строк, что нужно удалить. Также в глобальные переменные выделим константу для размерности массива строки: const int LENGTH = 255;
3. С клавиатуры считаем размерность массива и передадим его в ф-ию create\_matrix для двумерного массива, а далее в fill\_matrix – для заполнения двумерного массива;
4. Считаем k и проверяем, что n > k, если выполняется, то создаем новый двумерный массив, который имеет n-k строк, начиная с k-элемента через цикл: for (int i = 0; i < n - k; ++i) new\_rows[i] = rows[k + i].
5. Выводим матрицу: show\_matrix(new\_rows, n - k);
6. Удаляем память из под нового двумерного массива: delete[] new\_rows;
7. Удаляем память из под основной матрицы, а также ее элементов через ф-ию: delete\_matrix(rows, n);

**Алгоритм программы**

****







**Код программы:**

#include <iostream>

#include <windows.h>

using namespace std;

const int LENGTH = 255;

char\*\* create\_matrix(unsigned int n)

{

char\*\* rows = new char\*[n];

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

{

rows[i] = new char[LENGTH];

}

return rows;

}

void fill\_matrix(char\*\* rows, unsigned int n)

{

cout << "Введите " << n << " строк: " << endl;

cin.get();

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

{

cin.getline(rows[i], LENGTH);

}

}

void show\_matrix(char\*\* rows, unsigned int n)

{

cout << "Вывод " << n << " строк: " << endl;

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

{

cout << rows[i] << endl;

}

}

void delete\_matrix(char\*\* rows, unsigned int n)

{

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

{

delete[] rows[i];

}

delete[] rows;

}

char\*\* change\_matrix(char\*\* rows, unsigned int n, unsigned int k)

{

char\*\* new\_rows = create\_matrix(n - k);

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

{

new\_rows[i] = rows[k + i];

}

return new\_rows;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setlocale(LC\_ALL, "Rus");

unsigned int n, k;

cout << "Введите размерность матрицы: ";

cin >> n;

char\*\* rows = create\_matrix(n);

fill\_matrix(rows, n);

show\_matrix(rows, n);

cout << "Введите сколько строк удалить (n > k): ";

cin >> k;

if (k > n)

{

cout << "k > n" << endl;

}

else

{

char\*\* new\_rows = change\_matrix(rows, n, k);

show\_matrix(new\_rows, n - k);

delete[] new\_rows;

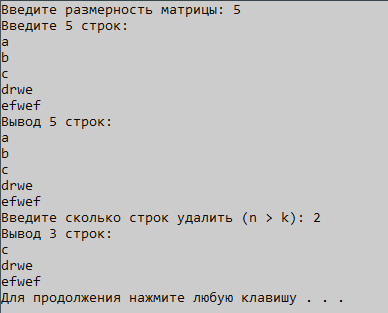
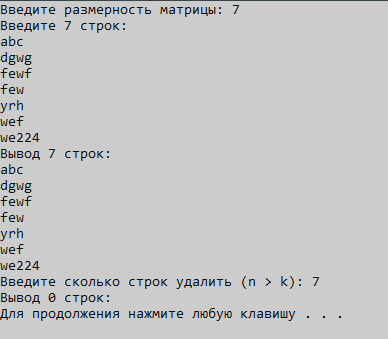
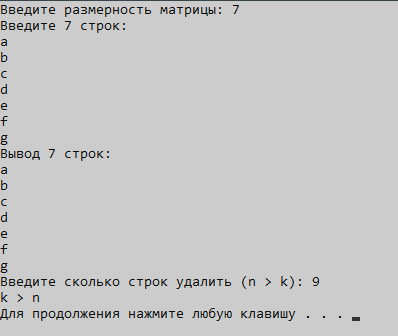
}

delete\_matrix(rows, n);

system("pause");

return 0;

}

**Скриншоты результатов:**