**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

# Лабораторная работа №3

по дисциплине «Программирование»

Массивы указателей

Группа: **АВТ-008**

Студент: **Казанцев К.О. , Павлюк А.С.**

Преподаватель: **Гриньков Д.К.**

НОВОСИБИРСК 2021

## Задание

1.      Функция получает линейный массив целых, находит в нем последовательности подряд возрастающих значений и возвращает их в динамическом массиве указателей на линейные массивы (аналог двумерного массива). В каждом из линейных динамических массивов содержится копия возрастающей последовательности, начиная с индекса 1, а под индексом 0 содержится его длина. Невозрастающие значения включаются в отдельный массив, добавляемый в конец (или начало) массива указателей.

## Проектирование программы

Обсуждение основных идей алгоритма

Идея: создать 4 динамических массива, 2 двумерных и 2 одномерных. Их нам хватит,

## Составные части программы

## 1. Создание двумерного динамического массива для буфера основного

|  |
| --- |
|  |
| int addlinedm(int\*\* &p, int k, int l, int d) { |
|  | int \*\*p1 = NULL; |
|  | p1 = new int\* [k]; //создаем буферный двумерный массив |
|  | for (int j = 0; j < k; j++) p1[j] = new int[d]; |
|  |  |
|  | for (int j1 = 0; j1 < k; j1++) //копируем значения в буферный массив |
|  | { |
|  | for (int j2 = 0; j2 < p[j1][0] + 1; j2++) { |
|  | p1[j1][j2] = p[j1][j2]; |
|  | } |
|  | } |
|  | k++; |
|  |  |
|  | for (int j = 0; j < k - 1; j++) delete[]p[j]; //удаление двумерного массива |
|  | delete[]p; |
|  |  |
|  | p = new int\* [k]; //пересоздаем основной массив |
|  | for (int j = 0; j < k - 1; j++) p[j] = new int[p1[j][0] + 1]; |
|  | p[k - 1] = new int[l + 1]; |
|  |  |
|  | for (int j1 = 0; j1 < k - 1; j1++) //копируем значения в основной массив |
|  | { |
|  | for (int j2 = 0; j2 < p1[j1][0] + 1; j2++) { |
|  | p[j1][j2] = p1[j1][j2]; |
|  | } |
|  | } |
|  |  |
|  | for (int j = 0; j < k - 1; j++) delete[]p1[j]; //удаление двумерного массива |
|  | delete[]p1; |
|  | return k; |
|  | } |
|  |  |

## 2. Создание массива для оставшейся невозрастающей последовательности

|  |
| --- |
| void addlinem(int\* &q) { |
|  | int\* q1 = NULL; |
|  | q1 = new int[q[0] + 1]; |
|  | for (int j = 0; j < q[0] + 1; j++) q1[j] = q[j]; |
|  | q1[0]++; |
|  | delete[]q; |
|  | q = new int[q1[0] + 1]; |
|  | for (int j = 0; j < q1[0]; j++) q[j] = q1[j]; |
|  | delete[]q1; |
|  | } |

## 3. Цикл заполнения двумерного массива

|  |
| --- |
| for (int i = 0; i < N-1; i++) |
|  | { |
|  | if (A[i] < A[i + 1]) { //считает возрастающую последовательность |
|  | v = 0; |
|  | l++; |
|  | } |
|  | else if (v==0){ //при конце возрастающей последовательности |
|  | v = 1; |
|  | if (l + 1 > d) d = l + 1; |
|  | k = addlinedm(p, k, l, d); |
|  |  |
|  | p[k-1][0] = l; |
|  | for (int j = 1; j < l + 1; j++) |
|  | { |
|  | p[k-1][j] = A[i - l + j]; |
|  | } |
|  | l = 1; |
|  | } |
|  | else if(v == 1) { //массив невозрастающих последовательностей |
|  | addlinem(q); |
|  | q[q[0]] = A[i]; |
|  | } |
|  | } |

## Вывод

## Программа верно распределяет последовательности в двумерный массив при помощи массивов указателей. Недостаток – буферный массив это матрица, а не массив массивов, при этом некоторые ячейки не используются.