

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Кафедра вычислительной техники

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ № 8
Одномерные массивы
Вариант № 12

Преподаватель

подпись, дата

Пушкарев К. В.

Студент КИ18-096, 031830645

подпись

05.11.2018

Котов С.А.

Красноярск 2018

1 Дополнительное упражнение № 2

1. Разработать, отладить, продемонстрировать и защитить преподавателю графическую схему алгоритма и составить программу, использующую динамическое выделение памяти под массив, для решения следующей задачи: из заданного одномерного массива $A[N]$ формирует новый массив B по правилам. Если массив сформировать не возможно, вывести соответствующее сообщение.

Задание 2: из всех элементов заданного массива за исключением последнего отрицательного.

Дополнительные указания 2: поиск индекса последнего отрицательного элемента массива оформить как функцию.

2. Провести трассировку программы с помощью встроенного в среду программирования отладчика, анализируя значения переменных после каждого оператора присваивания.

3. Выполнить несколько запусков программы для заранее подготовленных тестовых примеров для проверки работоспособности всех ветвей программы. Полученные результаты проанализировать.

2 Цель работы

Получить навыки построения графических схем алгоритма и программ обработки одномерных массивов; познакомиться с механизмом динамического выделения памяти.

3 Графическая схема алгоритма

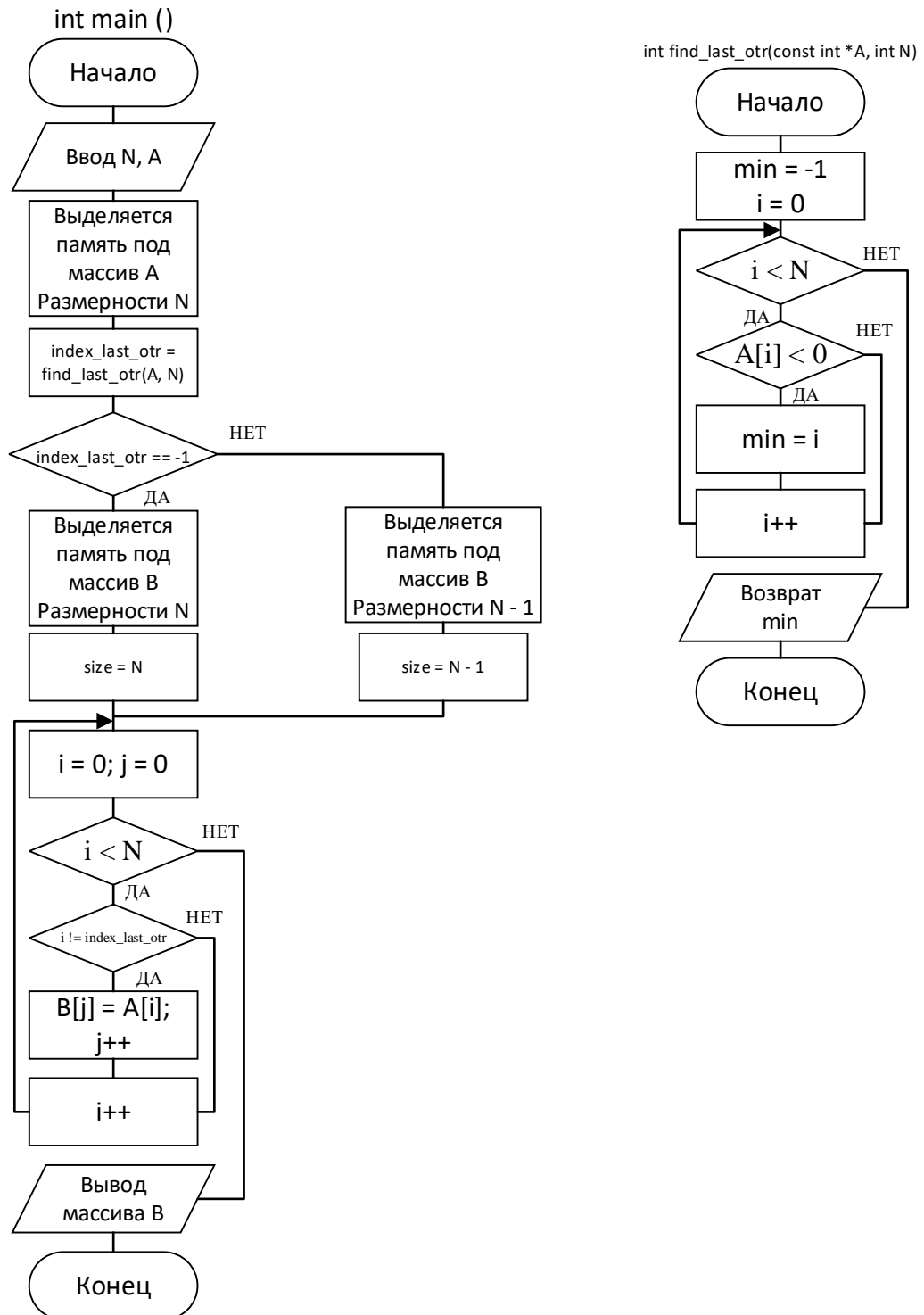


Рисунок 1 – Блок-схема дополнительного упражнения № 2.

4 Код программы

```

1 #include "pch.h"
2 #include <iostream>
3 #include <locale>
4
5 using namespace std;
6
  
```

```

7 int find_last_otr(const int *A, int N) {
8     int min = -1;
9     for (int i = 0; i < N; i++) {
10         if (A[i] < 0) {
11             min = i;
12         }
13     }
14     return min;
15 }
16
17 int main() {
18     setlocale(LC_ALL, "");
19
20     int N;
21     cout << "Введите значение N: ";
22     cin >> N;
23     int *A = new int[N];
24
25     cout << "Заполните массив A: ";
26     for (int i = 0; i < N; i++) {
27         cin >> A[i];
28     }
29
30     int index_last_otr = find_last_otr(A, N);
31     int size, *B;
32     if (index_last_otr == -1) {
33         B = new int[N];
34         size = N;
35     }
36     else {
37         B = new int[N - 1];
38         size = N - 1;
39     }
40
41     for (int i = 0, j = 0; i < N; i++) {
42         if (i != index_last_otr) {
43             B[j] = A[i];
44             j++;
45         }
46     }
47     cout << "Массив B : ";
48     for (int i = 0; i < size; i++) {
49         cout << B[i] << " ";
50     }
51
52     return 0;
53 }

```

5 Результат выполнения экспериментальной части работы.

Результаты запуска программы с различными входными значениями приведены в таблице 1.

Таблица 1 – Работа программы на тестовых наборах данных

Входные данные	Результат
h = 5 A = {1, 2, 3, 4, 5}	Массив B : 1 2 3 4 5

Входные данные	Результат
$h = 5$ $A = \{1, 2, -1, 5, -2\}$	Массив В : 1 2 -1 5