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 \_\_\_\_\_\_\_\_ Пушкарев К. В.

подпись, дата

Студент КИ18-09б, 031830645 \_\_\_\_\_\_ 05.11.2018 Котов С.А.

подпись

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  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53 | #include "pch.h"  #include <iostream>  #include <clocale>  using namespace std;  int find\_last\_otr(const int \*A, int N) {  int min = -1;  for (int i = 0; i < N; i++) {  if (A[i] < 0) {  min = i;  }  }  return min;  }  int main() {  setlocale(LC\_ALL, "");  int N;  cout << "Введите значение N: ";  cin >> N;  int \*A = new int[N];  cout << "Заполните массив A: ";  for (int i = 0; i < N; i++) {  cin >> A[i];  }  int index\_last\_otr = find\_last\_otr(A, N);  int size, \*B;  if (index\_last\_otr == -1) {  B = new int[N];  size = N;  }  else {  B = new int[N - 1];  size = N - 1;  }  for (int i = 0, j = 0; i < N; i++) {  if (i != index\_last\_otr) {  B[j] = A[i];  j++;  }  }  cout << "Массив B : ";  for (int i = 0; i < size; i++) {  cout << B[i] << " ";  }  return 0;  } |

**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 |
| h = 5  A = {1, 2, -1, 5, -2} | Массив B : 1 2 -1 5 |