**Лабораторная работа 1 (одномерные массивы)**

Содержание отчета:

1) Титульный лист

2) на отдельной странице задание лабораторной работы

3) исходный код программы

**замечание. В самом начале главного файла текст задания. Проделать все варианты!!!!**

Вариант A1

В массиве все четные элементы увеличить в 2 раза.

Пример: из массива A[5]: 1 3 4 5 6 должен получиться массив 1 3 8 5 12.

Вариант A2

В массиве все нечетные элементы увеличить на 10.

Пример: из массива A[5]: 1 3 4 5 6 должен получиться массив 1 13 4 15 6.

Вариант A3

В массиве все элементы, правее которых стоит нечетное значение, заменить на 0.

Пример: из массива A[5]: 1 3 4 5 6 должен получиться массив 0 3 0 5 6.

Вариант A4

В массиве все элементы, правее которых стоит четный элемент, уменьшить на 2.

Пример: из массива A[5]: 1 3 4 5 6 должен получиться массив 1 1 4 3 6.

Вариант A5

В массиве все четные элементы, стоящие между нечетными, уменьшить в 2 раза.

Пример: из массива A[5]: 1 3 4 5 6 должен получиться массив 1 3 2 5 6.

Вариант С1

Из массива удалить самую длинную цепочку четных элементов.

Пример: из массива A[8]: 4 1 4 2 1 2 4 6 должен получиться массив A[5]: 4 1 4 2 1.

(Самая длинная цепочка четных чисел включает элементы с 6 по 8: 2 4 6).

Вариант С2

Из массива удалить цепочки из четных элементов, состоящие менее чем из трех элементов.

Пример: из массива A[8]: 4 3 4 2 1 2 4 6 должен получиться массив A[5]: 3 1 2 4 6.

Вариант С3

Из массива удалить цепочки из нечетных элементов, состоящие менее чем из трех элементов.

Пример: из массива A[8]: 3 3 4 5 2 3 7 9 должен получиться массив A[5]: 4 2 3 7 9.

Вариант С4

Из массива A удалить те элементы, которые встречаются

и в массиве A, и в массиве B по крайней мере по 2 раза.

Пример: массив A[8]: 3 3 4 5 2 3 5 9

массив B[7]: 1 2 3 4 5 2 5.

По 2 раза в обоих массивах встречается только элемент, равный 5.

Массив A после удаления примет вид: A[6]: 3 3 4 2 3 9.

Вариант С5

Из массива из каждой цепочки четных элементов удалить

самый маленький элемент.

Пример: из массива A[9]: 3 6 4 5 2 3 4 6 4 должен

получиться массив A[6]: 3 6 5 3 6 4.

**Пример**

//Вычислить сумму тех элементов, значение которых больше значения их индекса.

//Пример: для массива A[5]: 2 1 3 7 4 сумма элементов равна 9.

#include "stdafx.h"

//этот модуль необходим, что бы быть использовать утверждения

#include <assert.h>

//этот модуль необходим, что бы можно было пользоваться функцией \_getch();

#include <conio.h>

//эти строки необходимы для того, что бы можно было пользоваться cint и cout

#include <iostream>

using namespace std;

const int MAXSIZE = 20;

int getSize() {

//выводим приглашение на ввод размера массива

cout << "input array size: ";

int size;

do {

//получаем из стандартного потока (т.е. клавиатуры) размер массива

cin >> size;

if (size <= 0) {

cout << "\nwrong size\n";

}

} while (size <= 0 || size > MAXSIZE);

return size;

}

void getArray(int \*a, const int size) {

//выводим приглашение на ввод элементов, указывая их количество

cout << "input " << size << " elements\n";

//собственно цикл по вводу элементов массива

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

cin >> a[i];

}

}

void finish() {

cout << "\n";

system("PAUSE");

}

int getSum(int \*a, int size) {

int sum = 0;

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

if (a[i] <= i + 1) {

continue;

}

sum += a[i];

}

return sum;

}

int \_tmain(int argc, \_TCHAR\* argv[]) {

//очищаем экран

system("cls");

//получаем размер массива

int size = getSize();;

//на всякий случай проверяем, что получили верно

assert(size >= 1);

assert(size <= MAXSIZE);

int a[MAXSIZE];

//получаем элементы массива

getArray(a, size);

cout << "sum is: " << getSum(a, size);

//выводим, что программа завершилась и ждем нажатия клавиши

finish();

return 0;

}

**Лабораторная работа 2 (статические массивы, знакомство с графическим режимом.)**

Общее задание:

В данной лабораторной работе требуется разработать программу, выполняющие следующие действия:

1. Вывод информации об авторе и задании в ГРАФИЧЕСКОМ РЕЖИМЕ. Так же необходимо нарисовать элементарную картинку, в которой продемонстрировано умение студента работать с графическим режимом

2. Ввод размера массива (максимальная размерность массива задается в программе заранее)

3. Ввод исходного массива (массивов)

4. Вывод введенных массивов

5. Обработка массива (массивов) в соответствии с вариантом

6. Вывод получившихся массивов

Замечания:

1) Количество элементов в исходных массивов не менее 20 штук.

2) Элементами массивов являются целые числа.

3) После каждого изменения массивов новое состояние необходимо вывести на экран.

4) "Скопировать элементы" - элементы из исходного массива добавляются в результирующий массив.

5) "Перенести элементы" - элементы из исходного массива добавляются в результирующий массив, после чего удаляются из исходного.

***Варианты задания***

1. Ввести массив А. В массив В скопировать все элементы массива А имеющие четный индекс и четное значение. Массив В отсортировать по убыванию, используя модифицированный метод простого выбора.

2. Ввести массив А. В массив В перенести все элементы массива А имеющие четный индекс и нечетное значение. Массив В отсортировать по возрастанию, используя модифицированный метод простого выбора.

3. Ввести массив А. В массив В скопировать все элементы массива А, имеющие четный индекс, слева от которых расположены элементы с нечетным значением. Массив В отсортировать по возрастанию, используя метод парных перестановок.