**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра ПИ**

Лабораторная работа № 10

По дисциплине «Основы алгоритмизации и программирования»

На тему «Одномерные массивы»

Выполнил:  
Студент 1 курса, 10 группа  
Короткевич Артём Сергеевич  
Преподаватель: асс. Харланович А.В

2024, Минск

**Задание 1**  
Выполнить программу, записанную в правой части. Опробовать второй вариант генерации чисел, записанный в комментарии.  
Добавить в программу операторы вычисления суммы элементов массива А.  
Произвести отладку.  
  
**Код**

#include <iostream>  
#include <ctime>

using namespace std;

int main() {

setlocale(LC\_ALL, "RU");

const int N = 100;

int i, sz, A[N], rmn = 0, rmx = 99, sum = 0;

cout << "Введите размер массива ";

cin >> sz;

cout << "Массив А:" << endl;

srand((unsigned)time(NULL));

for (i = 0; i < sz; i++) {

A[i] = (int)(((double)rand() / (double)RAND\_MAX) \* (rmx - rmn) + rmn);

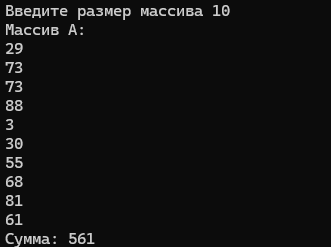
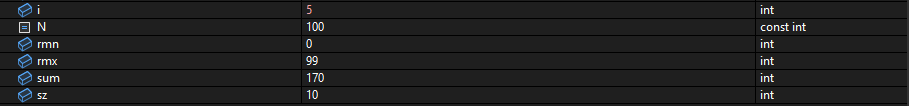
cout << A[i] << endl;

sum += A[i];

}

cout << "Сумма: " << sum;

}

**Результат  
  
  
  
  
  
  
  
  
Задание 2**Выполнить программу, записанную в правой части. Записать ее условие.  
Добавить в программу операторы вычисления среднего значения элементов исходного массива.  
Произвести отладку.  
  
**Код**

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

int i, k, size = 4;

float massivA[] = {5, -4, 17.1, 9, 1};

int sum = 0;

cout << "Введите номер элемента (от 0 до 4): ";

cin >> k;

for (i = k; i <= size; i++)

massivA[i] = massivA[i + 1];

size--;

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

{

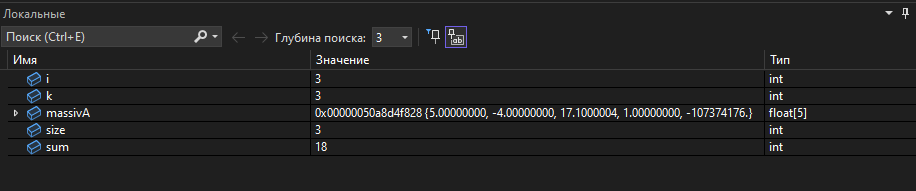
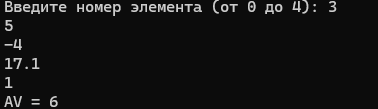
cout << massivA[i] << endl;

sum += massivA[i];

}

cout << "AV = " << sum / size;

}

**Результат  
  
  
  
  
  
  
  
  
  
  
  
Задание 3**Выполнить программу, приведенную в правой части.  
Внести изменения с тем, чтобы вычислялся минимальный элемент массива.  
Произвести отладку.  
  
**Код**

#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_ALL, "RU");

const int maxSize = 30;

int n, i, kmax = 0, kmin = 0;

int a[maxSize];

cout << "Введите размер массива:";

cin >> n;

if (n > 30)

return 0;

srand((unsigned)time(NULL));

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

a[i] = rand() % 30;

cout << a[i] << " ";

}

cout << endl;

for (i = 1; i < n; i++)

{

if (a[i] > a[kmax])

{

kmax = i;

}

}

cout << "Max: " << a[kmax] << endl;

for (i = 1; i != n; ++i)

{

if (a[i] < a[kmin])

{

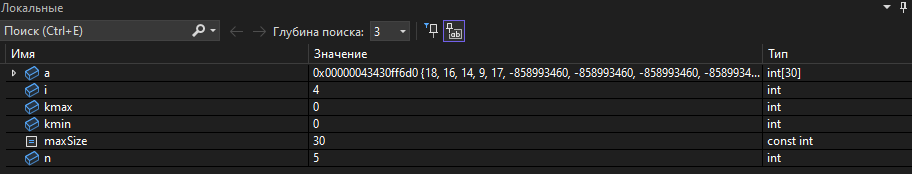
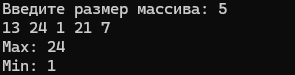
kmin = i;

}

}

cout << "Min: " << a[kmin];

}

**Результат  
  
  
  
  
  
  
  
  
  
  
  
Задание 4**Выполнить прокрутку программы, приведенной в правой части. Записать условие.

**Код**

#include <stdio.h>  
void main()

{

const int size = 5;

int masA[size], masB[size], masC[size \* 2];

int k = 0, j = 0, i = 0;

printf("A:\n");

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

scanf\_s("%d", &masA[n]);

printf("B:\n");

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

scanf\_s("%d", &masB[n]);

do

{

if (masA[k] <= masB[j])

masC[i++] = masA[k++];

else

masC[i++] = masB[j++];

if (k == size)

for (; j < size; j++)

masC[i++] = masB[j];

if (j == size)

for (; k < size; k++)

masC[i++] = masA[k];

} while (i < size \* 2);

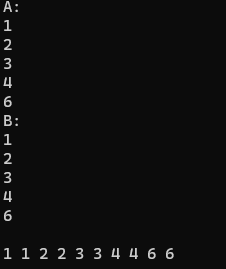
printf("\n");

for (i = 0; i < size \* 2; i++)

printf("%d ", masC[i]);

printf("\n");

}

**Прокрутка**Ввод элементов массива masA:  
Пользователь вводит, например: 1, 3, 5, 7, 9   
Ввод элементов массива masB:  
Пользователь вводит, например: 2, 4, 6, 8, 10   
Начало слияния массивов: k = 0, j = 0, i = 0   
Цикл do-while:   
Первая итерация:   
Сравниваем masA[0] = 1 и masB[0] = 2   
1 <= 2, поэтому masC[0] = 1  
 k = 1, i = 1   
Вторая итерация:   
Сравниваем masA[1] = 3 и masB[0] = 2   
3 > 2, поэтому masC[1] = 2  
 j = 1, i = 2   
Третья итерация:   
Сравниваем masA[1] = 3 и masB[1] = 4   
3 <= 4, поэтому masC[2] = 3   
k = 2, i = 3   
Четвертая итерация:   
Сравниваем masA[2] = 5 и masB[1] = 4   
5 > 4, поэтому masC[3] = 4  
 j = 2, i = 4   
Продолжаем аналогично до i < 10.  
Заполнение оставшихся элементов:  
Если k == size, копируем оставшиеся из masB.   
Если j == size, копируем оставшиеся из masA.  
Вывод массива masC: Результат: 1 2 3 4 5 6 7 8 9 10  
 **Результат  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Задание 5**В соответствии со своим вариантом написать программы по условиям, представленным в таблице ниже. Использовать одномерный массив целых случайных чисел (диапазон от 0 до 99). Размер массива ввести с клавиатуры.

Представить результаты в окне Отладчика.  
  
**Вариант 8**

1. Найти максимальный элемент массива и заменить им нечетные по номеру элементы.

2. Задан массив из **k** чисел. Определить количество инверсий в массиве (т. е. таких пар элементов, в которых большее число находится слева от меньшего).  
  
  
  
  
  
  
  
  
  
  
**Код(1)**

#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

int arr[] = {3, 7, 2, 8, 1, 9, 4};

int n = sizeof(arr) / sizeof(arr[0]);

int maxElement = arr[0];

for (int i = 1; i < n; ++i) {

if (arr[i] > maxElement) {

maxElement = arr[i];

}

}

for (int i = 1; i < n; i += 2) {

arr[i] = maxElement;

}

cout << "Изменённый массив = ";

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

cout << arr[i] << " ";

}

cout << endl;

return 0;  
}

**Результат(1)  
  
  
  
  
  
Код(2)**

#include <iostream>  
using namespace std;

int main() {

setlocale(LC\_CTYPE, "Russian");

int arr[] = { 3, 7, 2, 8, 1, 9, 4 };

int k = sizeof(arr) / sizeof(arr[0]); // размер массива

int сount = 0; // счёт инверсий

for (int i = 0; i < k - 1; ++i) { // перебор все пары элементов

for (int j = i + 1; j < k; ++j) // проверка, является ли пара инверсией

{

if (arr[i] > arr[j])

{

++сount;

}

}

}

cout << "Количество инверсий: " << сount << endl;   
 return 0;  
}

**Результат(2)  
  
  
  
  
  
Дополнительные задания**3. Найти в массиве наибольшее число подряд идущих одинаковых элементов.  
  
**Код**

#include <iostream>  
using namespace std;

int main()

{

setlocale(LC\_ALL, "RU");

const int n = 10;

int element[n];

for (int i = 0; i != n; ++i) {

cin >> element[i];

}

int counter = 1;

int maxCount = 0;

for (int i = 1; i != n; ++i)

{

if (element[i] == element[i - 1])

{

counter++;

if (counter > maxCount)

{

maxCount = counter;

}

}

else {

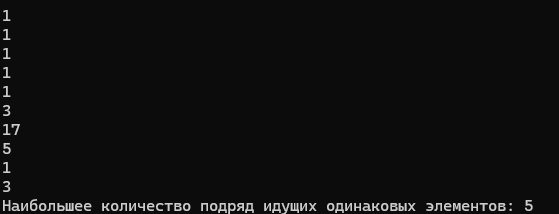
counter = 1;

}

}

cout << "Наибольшее количество подряд идущих одинаковых элементов: " << maxCount;

}

**Результат  
  
**