

**УВАЖАЕМЫЕ СТУДЕНТЫ!** Изучите теоретические сведения к лабораторной работе, выполните практическое задание.

Результаты работы, фотоотчет, предоставить преподавателю на e-mail: [r.bigangel@gmail.com](mailto:r.bigangel@gmail.com) **до 10.04.2023.**

**Требования к отчету:**

Отчет предоставляется преподавателю в электронном варианте и должен содержать:

- название работы, постановку цели, вывод;
- ответы на контрольные вопросы, указанные преподавателем.

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)111-37-59, (Viber, WhatsApp), vk.com: <https://vk.com/daykini>

***ВНИМАНИЕ!!!*** При отправке работы, не забывайте указывать *ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).*

### Лабораторная работа № 13

**Тема: «Селективная обработка элементов массива. Нахождение минимального и максимального элементов массива»**

**Цель:** изучить понятия, особенности внутреннего представления, способов генерации и вывода одномерных массивов, научиться выполнять объявление, инициализацию, генерацию и вывод массивов при решении задач на языке C++.

### Теоретическая часть

В языке программирования C заложены средства для задания последовательностей упорядоченных данных. Такие последовательности называются массивами. В массивах должны быть упорядочены данные одного и того же типа. В данной лабораторной работе будут рассматриваться массивы с целыми и вещественными типами данных, т.е. типы int, float или double.

Массивы данных могут быть одномерными (векторами размера  $1 \times n$  или  $n \times 1$ ), двумерными (матрицами размера  $n \times m$ ) или многомерными (размера  $n \times m \times p \dots$ ). В частности, для векторов и матриц в приведенной записи первый индекс означает количество строк, а второй (число или буква) – это количество столбцов. Для названия массива может быть использована переменная, состоящая из букв (буквы), букв с цифрами, букв с цифрами и знаком подчеркивания и т.д. в соответствии с правилами объявления

переменных, принятых в языке С. Если размерность массива меньше, чем требуется, то компилятор не выдаст сообщения об ошибке. Выход за границы массивов должен следить только сам программист.

## 1. Одномерные массивы

Одномерный массив – это список связанных однотипных переменных.

Общая форма записи одномерного массива:

```
тип имя_массива [размер] ;
```

В приведенной записи элемент тип объявляет базовый тип массива. Количество элементов, которые будут храниться в массиве с именем имя\_массива, определяется элементом размер.

В языке С индексация массива начинается с нуля. Например, если размер массива определен величиной 9, то в массиве можно хранить 10 элементов с индексацией 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Доступ к отдельному элементу массива осуществляется с помощью индекса. Индекс описывает позицию элемента внутри массива.

Все массивы занимают смежные ячейки памяти, т.е. элементы массива в памяти расположены последовательно друг за другом. Ячейка памяти с наименьшим адресом относится к первому элементу массива, а с наибольшим – к последнему.

Для одномерных массивов общий размер массива в байтах вычисляется по формуле:

$$\text{всего байт} = \text{размер типа в байтах} * \text{количество элементов}$$

В языке С нельзя присвоить один массив другому. Для передачи элементов одного массива другому необходимо выполнить присвоение поэлементно.

## 2. Инициализация массива

В языке С++ массив при объявлении можно инициализировать.

Общая форма инициализации массива:

```
тип имя_массива [размер1] * [размерN] =  
{ список_значений } ;
```

В список\_значений входят константы, разделенных запятыми. Типы констант должны быть совместимыми с типом массива.

Пример инициализации одномерного массива:

```
int A[5] = {1, 2, 3, 4, 5};
```

При этом  $A[0] = 1$ ,  $A[1] = 2$  и т.д.

В языке C возможна инициализация безразмерных массивов.

Например, для одномерного массива:

```
int A[ ] = {1, 2, 3, 4, 5};
```

### Практическая часть

**Пример 1.** *Напишите программу заполнения одномерного массива случайными числами из интервала от 1 до 15 по случайному равномерному закону. Отсортировать массив случайных чисел по возрастанию.*

Для решения поставленной задачи применим сортировку методом прямого выбора. Алгоритм сортировки заключается в следующем:

1. В исходной последовательности из  $N$  элементов отыскивается элемент с наименьшим ключом.
2. Он меняется местами с первым элементом.
3. В оставшейся последовательности из  $(N-1)$  элементов отыскивается минимальный элемент и меняется местами со вторым элементом и т.д., пока не останется один, самый большой элемент.

Программный код решения примера:

```
#include <stdio.h>
#include <conio.h>
#include <time.h>
#include <stdlib.h>
#define Left 1
#define Right 15
#define N 10

int main (void) {
float R, r, min;
float A[N];

int i, j, k;
unsigned int some;
long int L;

L = (long) time(NULL); // Системное время
some = (unsigned) L; // Приведение типов
srand(some); // Задание исходного случайного числа для
rand()
printf("\n\t The initial array of random numbers in the
interval [%d, %2d]\n", Left, Right);

for (i = 0; i < N; ++i)
{ // Случайное число из интервала [0,1]
r = (float) rand()/RAND_MAX;
```

```

// Формирование случайного числа из заданного интервала
R = Left + (Right - Left) * r;
// Заполнение массива случайными числами
A[i] = R; }

// Печать элементов исходного массива
for (i = 0; i < N; ++i)
printf("\n\t %5d) %10.4f", i + 1, A[i]);

// Сортировка методом выбора
for (i = 0; i < (N - 1); ++i)
{
    min = A[i]; k = i;
    for (j = i + 1; j < N; ++j)
    if (A[j] < min) { k = j; min = A[k]; }
    A[k] = A[i]; A[i] = min;
}

// Печать отсортированного массива по возрастанию
printf("\n\n\t Sort an array:\n");
for (i = 0; i < N; ++i)
printf("\n\t %5d) %10.4f", i + 1, A[i]);

    printf("\n\n Press any key: ");
    _getch();
    return 0;
}

```

Возможный результат выполнения программы показан на рис. 5.1.

```

e:\Projects_C\Lab5\Debug\Lab5.exe
The initial array of random numbers in the interval [1, 15]
1) 10.9855
2) 13.6627
3) 8.5642
4) 4.6355
5) 14.3651
6) 9.5221
7) 1.4862
8) 4.8769
9) 7.0799
10) 7.6652

Sort an array:
1) 1.4862
2) 4.6355
3) 4.8769
4) 7.0799
5) 7.6652
6) 8.5642
7) 9.5221
8) 10.9855
9) 13.6627
10) 14.3651

Press any key: _

```

**Рис. 1.** Сортировка одномерного массива по возрастанию

В программе использованы директивы препроцессора для задания левой границы (`#define Left 1`), правой границы (`#define Right 15`) и размера одномерного массива (`#define N 10`). Включены дополнительные библиотеки **time.h** – для обращения к функциям системного времени, **stdlib.h** – для обращения к функциям генерации псевдослучайных чисел.

**Пример 2.** *Напишите программу поиска максимального элемента в заданном одномерном массиве. Элементы массива являются целыми числами.*

Программный код решения примера:

```

#include <stdio.h>
#include <conio.h>

int main (void){
    int i, size, max;
    int A[ ] = {3, 5, 2, 8, 12, 0, -7, -3, -21};
    size = sizeof(A)/sizeof(A[0]);
    printf("\n\t The dimention of the array A is equal to:
%d\n", size);
    max = A[0]; // Предполагаемый максимум
    for (i = 0; i < size; ++i)
        if (A[i] > max) max = A[i];

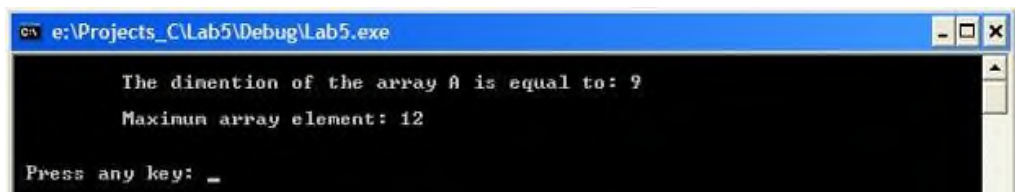
    printf("\n\t Maximum array element: %d\n", max);

    printf("\n\n Press any key: ");
    _getch();
    return 0; }

```

В программе использована инициализация безразмерного массива и определения его размерности с помощью функции sizeof().

Результат выполнения программы показан на рис. 5.2.



```

e:\Projects_C\Lab5\Debug\Lab5.exe
The dimention of the array A is equal to: 9
Maximun array element: 12
Press any key: _

```

**Рис. 2.** Определение максимального элемента массива

**Пример 3.** *Напишите программу циклической перестановки чисел заданного массива так, чтобы  $i$ -е число стало  $(i+1)$ -м, а последнее число – первым. Выведите на дисплей исходный массив и преобразованный.*

Программный код решения примера:

```

#include <stdio.h>
#include <conio.h>
#define N 55

int main (void) {
    int i, j, k;
    double D[ ] = {1.23, 2.34, 3.45, 4.56, 5.67,
6.78};
    double B[N]; // Заведомо больший размер, чем у
массива D

```

```

// Обнуление массива и выделение памяти для него
for (i = 0; i < N; ++i)
    B[i] = 0.0;

k = sizeof(D)/sizeof(D[0]);

B[0] = D[k-1];
for (i = 0; i < (k - 1); ++i)
    B[i+1] = D[i];

printf("\n\t The original array:\n");
for (i = 0; i < k; ++i)
    printf("%8.2f", D[i]);

printf("\n\n\t The reconfigured array:\n");
for (j = 0; j < k; ++j)
    printf("%8.2f", B[j]);

    printf("\n\n Press any key: ");
    _getch();
    return 0;
}

```

Результат выполнения программы показан на рис. 5.3.

```

e:\Projects_C\Lab5\Debug\Lab5.exe
    The original array:
  1.23   2.34   3.45   4.56   5.67   6.78
    The reconfigured array:
  6.78   1.23   2.34   3.45   4.56   5.67
    Press any key: _

```

**Рис. 3.** Пример циклической перестановки элементов числового массива

**Пример 4.** В данном одномерном массиве вещественных чисел поменяйте местами элементы, стоящие на нечетных местах, с элементами, стоящими на четных местах. Предусмотрите четность и нечетность размерности массива.

Для определения четности места в заданном массиве можно использовать операцию деления по модулю, т.е. `%`.

Программный код решения примера:

```

#include <stdio.h>
#include <conio.h>
// Размер массива
#define n 7

```

```

int main (void) {
int i, k;
// Пример массива
float A[n] = {1.23F, 2.34F, 3.45F, 4.56F, 5.67F, 6.78F,
7.89F};
float B[n]; // Вспомогательный массив
// Обнуление массива
    for (i = 0; i < n; ++i)
        B[i] = 0;
// Распечатка заданного массива
printf("\n\t\t The original array of dimation n =
%d:\n", n);
    printf("\t");
    for (i = 0; i < n; ++i)
        printf("%6.2f", A[i]);
// Распечатка преобразованного массива
    printf("\n\n\t\t The reconfigured array:\n");
    for (i = 0; i < n; ++i) {
k = i % 2; // Для определения четности индекса массива
if (k == 0 && i < n - 1 )
    B[i] = A[i + 1];
else if (k != 0 && i > 0 )
B[i] = A[i-1];
else if (k == 0 && i < n)
B[i] = A[i]; } printf("\t");
        for (i = 0; i < n; ++i)
            printf("%6.2f", B[i]);
printf("\n\n Press any key: ");
    _getch();
return 0; }

```

При инициализации массива каждый его элемент снабжен суффиксом **F**.

Результат выполнения программы показан на рис. 5.4.

```

e:\Projects_C\Lab5\Debug\Lab5.exe
    The original array of dimation n = 7:
1.23 2.34 3.45 4.56 5.67 6.78 7.89
    The reconfigured array:
2.34 1.23 4.56 3.45 6.78 5.67 7.89
Press any key: _

```

**Рис. 4.** Смена четных и нечетных мест чисел массива

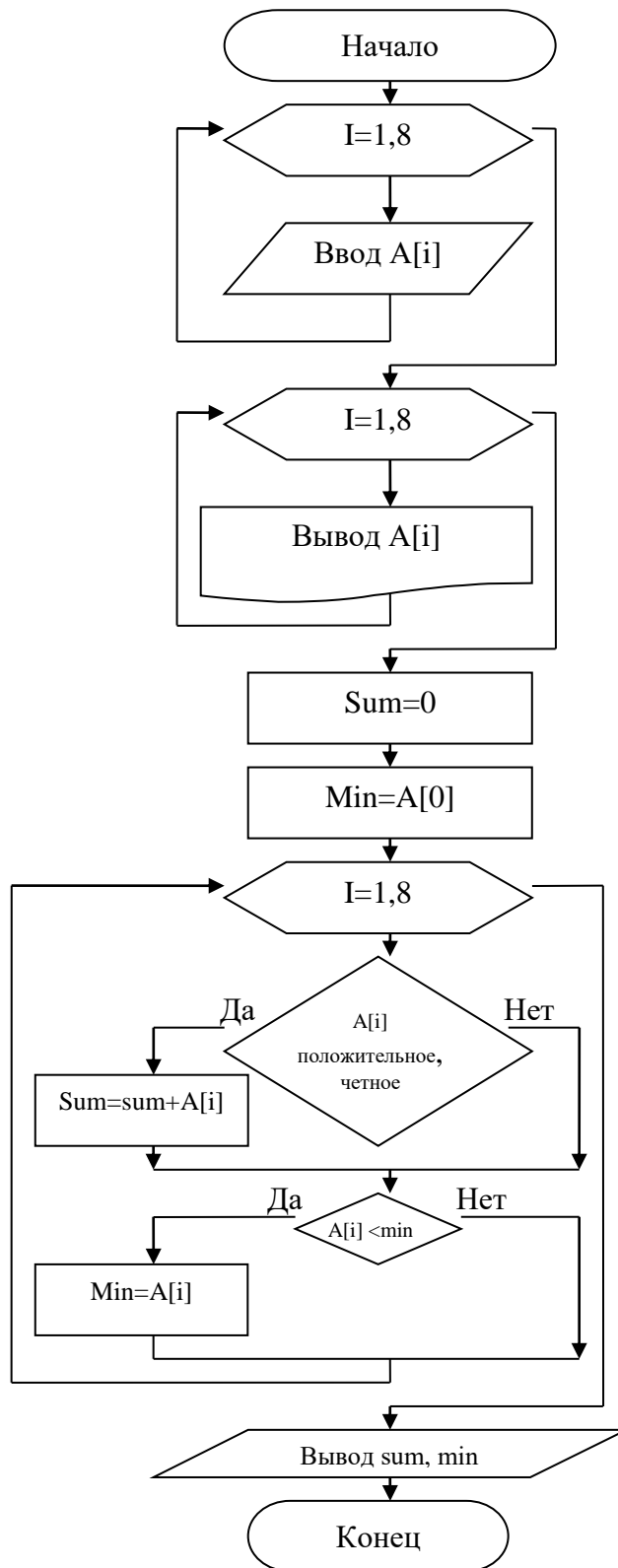


**Пример 5.** Создать массив  $A(8)$ , элементы которого ввести с клавиатуры, найти сумму четных положительных элементов и минимальный элемент массива.

Программный код решения примера:

```
// Пятая программа на языке Си++
#include <stdio.h>
#include <conio.h>
#include <math.h>
// #include <stdlib.h>
int main (void) {
int A[8];
int i, sum, min;
printf("\n\t Vvedite massiv iz 8 chisel\n");
for (i = 0; i < 8; i++)
scanf("%i", &A[i]);
// Печать элементов исходного массива
printf("\n\t ishodniy massiv \n");
for (int i = 0; i < 8; i++)
printf("\n\t %5d element massiva %10d", i + 1, A[i]);
// Нахождение суммы четных положительных элементов
sum=0; min=A[0];
for (i = 0; i < 8; i++)
    { if((A[i]>0)&&!(A[i]%2)) {sum=sum+A[i];}
      if (A[i] < min) {min = A[i];}}
printf("\n\t Minimalniy elementmassiva %d\n", min);
// Печать найденной суммы
printf("\n\n\t Summa=:%10d\n", sum);
printf("\n\n Press any key: ");
_getch();
return 0;
}
```

*Блок-схема программы*



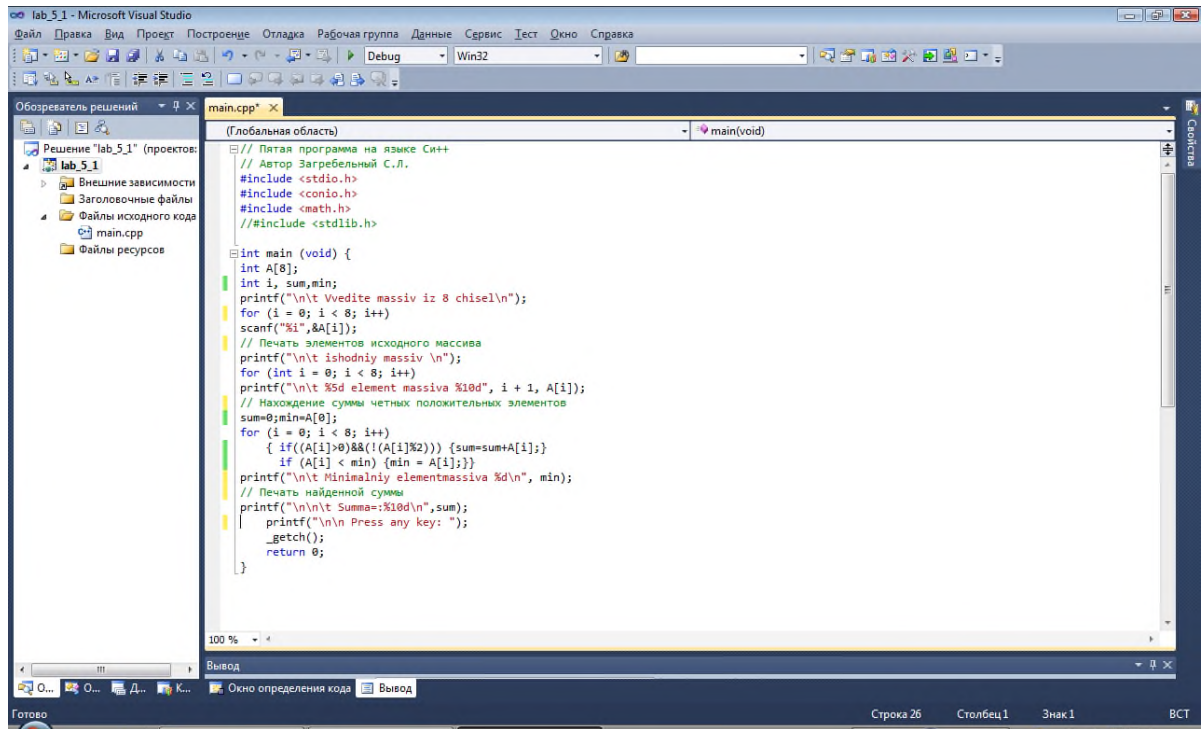


Рис. 5 Окно кода программы

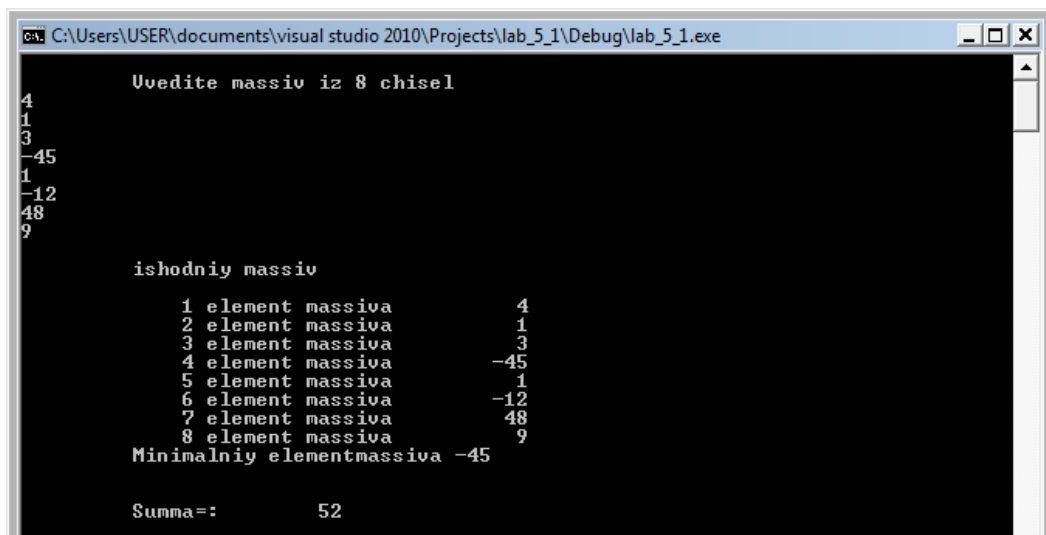


Рис. 6. Окно обработки элементов массива

## Индивидуальные задания

Составить блок-схемы и программы для решения следующих задач  
таблица 1.

**Таблица 1.**

Вариант	задание	Условие задания
1	<i>a</i> <i>б</i> <i>в</i>	Найти количество положительных элементов Найти сумму элементов больших 3 Найти максимальный элемент массива
2	<i>a</i> <i>б</i> <i>в</i>	Найти количество отрицательных элементов Найти сумму отрицательных элементов Найти минимальный элемент кратный пяти
3	<i>a</i> <i>б</i> <i>в</i>	Найти количество четных элементов Найти сумму элементов кратных 3 Найти разность максимального и минимального элементов массива
4	<i>a</i> <i>б</i> <i>в</i>	Найти среднее арифметическое элементов массива Найти сумму наибольшего и наименьшего элементов массива Найти максимальный по модулю элемент массива
5	<i>a</i> <i>б</i> <i>в</i>	Вычислить среднее арифметическое четных элементов массива Найти сумму минимального положительного элемента массива и его номера Найти произведение модулей наибольшего отрицательного и наименьшего четного элементов массива
6	<i>a</i> <i>б</i> <i>в</i>	Найти количество элементов кратных 5 Найти сумму четных элементов массива стоящих на нечетных местах Найти сумму второго и наибольшего положительного элементов массива
7	<i>a</i> <i>б</i> <i>в</i>	Найти среднее геометрическое четных элементов массива Найти номер наибольшего по модулю элемента массива Найти максимальный четный элемент массива
8	<i>a</i> <i>б</i>	Вычислить среднее арифметическое максимального и минимального элементов массива Найти минимальный по модулю элемент массива

	<i>в</i>	Найти сумму элементов из интервала [0;10]
<b>9</b>	<i>а</i>	Вычислить среднее геометрическое номеров максимального и минимального элементов массива
	<i>б</i>	Найти разность суммы положительных и произведения отрицательных чисел массива
	<i>в</i>	Найти количество положительных элементов
<b>10</b>	<i>а</i>	Найти сумму отрицательных элементов
	<i>б</i>	Найти сумму элементов массива, у которых индекс кратен 3
	<i>в</i>	Найти произведение модулей наибольшего и наименьшего элементов массива
<b>11</b>	<i>а</i>	Найти сумму отрицательных элементов
	<i>б</i>	Найти сумму второго и наибольшего положительного элементов массива
	<i>в</i>	Найти разность максимального и минимального элементов массива
<b>12</b>	<i>а</i>	Вычислить среднее арифметическое четных элементов массива
	<i>б</i>	Найти сумму элементов в диапазоне [-10;20]
	<i>в</i>	Найти максимальный по модулю элемент массива
<b>13</b>	<i>а</i>	Найти сумму минимального положительного элемента массива и его номера
	<i>б</i>	Найти сумму четных элементов массива из диапазона [-20;30]
	<i>в</i>	Найти минимальный по модулю элемент массива
<b>14</b>	<i>а</i>	Найти количество элементов кратных 5
	<i>б</i>	Найти сумму четных элементов массива, стоящих на нечетных местах
	<i>в</i>	Найти максимальный четный элемент массива
<b>15</b>	<i>а</i>	Найти количество положительных элементов
	<i>б</i>	Найти сумму элементов больших 3
	<i>в</i>	Найти максимальный элемент массива
<b>16</b>	<i>а</i>	Найти количество отрицательных элементов
	<i>б</i>	Найти сумму отрицательных элементов
	<i>в</i>	Найти минимальный элемент кратный пяти
<b>17</b>	<i>а</i>	Найти количество четных элементов
	<i>б</i>	Найти сумму элементов кратных 3
	<i>в</i>	Найти разность максимального и минимального элементов массива
<b>18</b>	<i>а</i>	Найти среднее арифметическое элементов массива
	<i>б</i>	Найти сумму наибольшего и наименьшего элементов массива

	<i>в</i>	Найти максимальный по модулю элемент массива
19	<i>а</i>	Вычислить среднее арифметическое четных элементов массива
	<i>б</i>	Найти сумму минимального положительного элемента массива и его номера
	<i>в</i>	Найти произведение модулей наибольшего отрицательного и наименьшего четного элементов массива
20	<i>а</i>	Найти количество элементов кратных 5
	<i>б</i>	Найти сумму четных элементов массива стоящих на нечетных местах
	<i>в</i>	Найти сумму второго и наибольшего положительного элементов массива
21	<i>а</i>	Найти среднее геометрическое четных элементов массива
	<i>б</i>	Найти номер наибольшего по модулю элемента массива
	<i>в</i>	Найти максимальный четный элемент массива
22	<i>а</i>	Вычислить среднее арифметическое максимального и минимального элементов массива
	<i>б</i>	Найти минимальный по модулю элемент массива
	<i>в</i>	Найти сумму элементов из интервала [0;10]
23	<i>а</i>	Вычислить среднее геометрическое номеров максимального и минимального элементов массива
	<i>б</i>	Найти разность суммы положительных и произведения отрицательных чисел массива
	<i>в</i>	Найти количество положительных элементов
24	<i>а</i>	Найти сумму отрицательных элементов
	<i>б</i>	Найти сумму элементов массива, у которых индекс кратен 3
	<i>в</i>	Найти произведение модулей наибольшего и наименьшего элементов массива
25	<i>а</i>	Найти сумму отрицательных элементов
	<i>б</i>	Найти сумму второго и наибольшего отрицательного элементов массива
	<i>в</i>	Найти разность максимального и минимального элементов массива
26	<i>а</i>	Вычислить среднее арифметическое четных элементов массива
	<i>б</i>	Найти сумму элементов в диапазоне [-10;20]
	<i>в</i>	Найти максимальный по модулю элемент массива
27	<i>а</i>	Найти сумму минимального положительного

	<i>б</i>	элемента массива и его номера
	<i>в</i>	Найти сумму четных элементов массива из диапазона [-20;30]
	<i>в</i>	Найти минимальный по модулю элемент массива
28	<i>а</i>	Максимальный по модулю элемент
	<i>б</i>	Найти среднее арифметическое элементов массива
	<i>в</i>	Найти сумму отрицательных элементов
29	<i>а</i>	Найти количество элементов кратных 4
	<i>б</i>	Найти сумму отрицательных элементов
	<i>в</i>	Найти сумму наибольшего и наименьшего элементов массива
30	<i>а</i>	Найти разность максимального и минимального положительных элементов
	<i>б</i>	Найти сумму нечетных элементов
	<i>в</i>	Найти минимальный элемент из диапазона [-20;30]

### Контрольные вопросы

1. Как организуются одномерные числовые массивы в языке C++?
2. Как организуется индексирование числовых массивов в языке C++?
3. На кого или на что возлагается контроль границ числовых массивов в языке программирования C++?
4. Для чего применяется начальная инициализация числовых массивов при дальнейшем их использовании?
5. Почему при определении размерности массива с помощью препроцессорной директивы **define** не используется точка с запятой после числового значения?