

**ЛАБОРАТОРНАЯ РАБОТА № 18
ДОПОЛНИТЕЛЬНЫЕ ЗАДАЧИ С МАССИВАМИ
И АЛГОРИТМЫ ИХ РЕШЕНИЯ**

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Задачи по работе с массивами можно разбить на следующие типы:

1. Последовательная обработка элементов массива.
2. Выборочная обработка элементов массива.
3. Поиск в массиве единственного элемента, отвечающего некоторому условию.
4. Одновременная обработка нескольких массивов.
5. Изменение порядка следования элементов без изменения размеров исходного массива.
6. Переформирование массива с изменением его размеров.

Для каждого типа существуют свои приемы программирования. В реальных программах задачи в чистом виде встречаются довольно редко, но любую сложную задачу можно разделить на более простые задачи, относящиеся к указанным выше типам.

Большинство задач было рассмотрено ранее в лабораторной работе 12.

Изменение порядка следования элементов без изменения размеров исходного массива.

Некоторые задачи данного типа (перестановка-инверсия и циклический сдвиг) уже рассматривались в лабораторной работе 12. Кроме них встречаются другие задачи, в которых изменяется порядок следования элементов (переформируется весь массив или его часть). Чаще всего при этом приходится менять местами значения двух элементов массива.

Самый простой алгоритм, позволяющий поменять местами значения двух переменных, использует дополнительную переменную. Например, значения переменных *a* и *b* после выполнения следующих команд поменяются:

```
int a = 4, b = 6;
int c;
c = a;    // a = 4; b = 6; c = 4;
a = b;    // a = 6; b = 6; c = 4;
b = c;    // a = 6; b = 4; c = 4;
```

Лучше всего для этих целей использовать функцию, описанную в лабораторной работе 11. Аргументы в нее передаются через указатели, поэтому значения переменных, на которые они будут указывать, поменяются местами.

```
void Swap(int *pa, int *pb) // функция перестановки
{
    int c = *pa;
    *pa = *pb;
    *pb = c;
}
```

При вызове функции в нее нужно передавать адреса переменных, значения которых требуется поменять.

```
int a = 4, b = 6;
swap(&a, &b);
```

Изменять порядок следования элементов приходится при сортировке массива или просто по некоторому условию, например, переформировать массив так, чтобы сначала в нем шли четные, а потом нечетные элементы, или сначала отрицательные элементы, а потом большие или равные нулю и т.д.

Пример 1. Переставить в начало массива все числа, которые меньше 0

Для решения данной задачи в цикле просматриваем все элементы с начала массива (индекс j) и ищем элемент, который нужно переставить в конец (т.е. тот, который неотрицательный). Затем просматриваем все элементы с конца массива (индекс k) и ищем элемент, который нужно переставить в начало (т.е. отрицательный). Меняем их местами. Цикл продолжается, пока индекс j меньше k .

```
int Arr[N];
int j = 0, k = N - 1;

while (j < k)
{
    if (Arr[j] < 0)
        j++;
    else if (Arr[k] >= 0)
        k--;
    else
        swap(&Arr[j], &Arr[k]);
}
```

Переформирование массива с изменением его размеров

К данному классу задач относятся: удаление из массива элементов, соответствующих некоторому условию (например, удаление всех нечетных элементов), вставка в массив дополнительных элементов в указанные места массива, выделение подмассива, объединение массивов и т.д. При этом может потребоваться изменить размер массива. Это можно сделать только для динамических массивов.

Пример 2. Удаление элементов массива

При удалении k -того элемента из массива (без изменения его размера) нужно просто сдвинуть элементы влево, начиная с номера $k + 1$:



```
for (i = k; i < N-1; i++) // цикл от k до N-2
    A[i] = A[i+1];
```

При удалении всех элементов с заданными свойствами из массива (без изменения его размера) нужно, просматривая в цикле все элементы, найти очередной элемент с заданными свойствами и удалить его в соответствии с вышеописанным алгоритмом (лучше сделать функцию). Количество элементов в массиве станет на 1 меньше, что можно учесть при выводе их на дисплей.

Например, чтобы удалить из массива все элементы, равные 0, напомним функцию удаления k -того элемента

```
void Delete(int *A, int Len, int k) // функция удаления k-того
{                                     // элемента
    for (int i = k; i < Len-1; i++)
        A[i] = A[i+1];
}
```

А затем в цикле, просматривая все элементы массива, ищем те, которые равны 0, и удаляем их.

```
for (int j = 0; j < N; j++)  
    if (Arr[j] == 0)  
        Delete(Arr, N, j);
```

Можно использовать другой алгоритм, в котором будет только один цикл. Используем два индекса: по индексу k пишем элемент массива с индексом i , если он не равен 0.

```
for (int i = 0, k = 0; i < Size; i++)  
{  
    if (A[i] != 0)  
    {  
        A[k] = A[i];  
        k++;  
    }  
}
```

Пример 3. Вставка дополнительных элементов массива

Для добавления в массив элемента (например, после элемента k) необходимо следить за размером массива. Если размер позволяет, нужно сдвинуть элементы вправо, начиная с номера $N-2$ и заканчивая $k+1$:

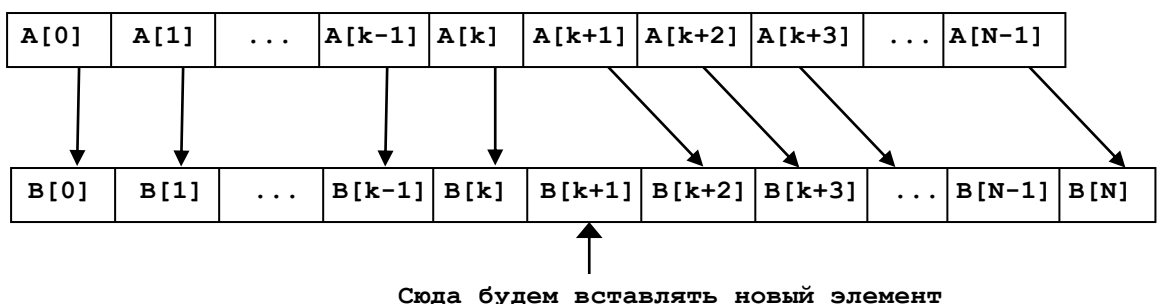


```
for (i = N - 2; i > k; i--)  
    A[i+1] = A[i];  
A[k+1] = newElem;    // Вставляем новый элемент после k-того
```

Лучше сделать функцию:

```
void AddElem(int* A, int Size, int k, int newElem)  
{  
    for (int i = Size - 2; i > k; i--)  
        A[i + 1] = A[i];  
    A[k + 1] = newElem;  
}
```

Если выходим за пределы массива, нужно увеличить его размер. Это можно сделать только для динамических массивов. Выделяем память под новый массив, запоминая его адрес во временной переменной, копируем в новый массив нужные переменные, освобождаем память, выделенную под старый массив, и запоминаем адрес в этом указателе:



Вставляем новый элемент со значением newElem после k -того:

```

int *A = new int[N];    // «старый» массив
// заполнение массива
. . . . .
int *B = new int[N+1];  // Выделяем память под новый массив
                        // размером N+1

for (i = 0; i <= k; i++) // Копируем от 0 до k включительно
    B[i] = A[i];
for (i = k + 1; i < N; i++) // Копируем от k+1 до N-1
    B[i+1] = A[i];         // со сдвигом

B[k+1] = newElem;    // Вставляем новый элемент после k-того
delete[] A;          // Освобождаем «старую» память
A = B;               // Запоминаем в «старом» указателе адрес нового
массива

```

Если нужно вставить в массив несколько дополнительных элементов, например после тех, элементов массива, которые равны 0, вставить -1, нужно сначала посчитать на сколько увеличится размер массива, затем создать новый массив нужного размера, а затем в него копируем элементы из старого массива, добавляя, если нужно, новые элементы.

```

int n = 10;
int *Arr = new int[n];
// Заполнение массива
. . . . .
int ntmp = n;
for(int i = 0; i < n; i++)    // Считаем кол-во элементов,
    if (Arr[i] == 0)         // которые нужно добавить
        ntmp++;

int* ArrTmp = new int[ntmp]; // Выделяем новую память
for (int i = 0; i < n; i++)   // Копируем из старого массива
    ArrTmp[i] = Arr[i];
n = ntmp; // Размер нового массива

for (int i = 0; i < n - 1; i++)
{
    if (ArrTmp[i] == 0) // Просматриваем все элементы,
    {                  // если нашли равный 0,
        AddElem(ArrTmp, Size, i, -1); // вставляем после него -1
        i++;
    }
}

delete[] Arr; // Освобождаем старую память
Arr = ArrTmp; // Запоминаем в указателе адрес нового массива

```

Выделение подмассива и объединения массивов выполняются аналогично.

Нахождение размаха, моды и медианы

Кроме минимального и максимального значений при решении различных задач с массивами приходится искать и другие характеристики, среднее арифметическое, среднее геометрическое, размах, моду и медиану.

Размах ряда чисел – это разность между наибольшим и наименьшим из этих чисел.

Мода ряда чисел – это число, которое встречается в данном ряду чаще других.

Ряд чисел может иметь более одной моды, а может не иметь моды совсем. Примеры: модой ряда 1, 7, 3, 8, 7, 2, 2, 7, 4, 9, 8 является число 7, встречается 3 раза. В ряду чисел 1, 2, 4, 6, 5, 8 моды нет. Ряд 1, 1, 2, 2, 3 содержит 2 моды: 1 и 2.

Для поиска моды необходим вспомогательный массив счёта (массив рейтинга) — это подобие самого обычного счётчика, но только для массива значений. Размер этого массива равен количеству возможных значений анализируемого массива. Например, если в нем используются только числа от 0 до 9, то размер вспомогательного массива равен 10, а если нужно проанализировать частоту повторений символов в тексте, то размер вспомогательного массива должен быть равен количеству символов в алфавите либо 256 (в таблице ASCII 256 символов). Каждая ячейка массива используется как счетчик для подсчета встречаемости соответствующего числа (или символа).

В цикле просматриваются все элементы анализируемого массива, если встреченное число 5, то во вспомогательном массиве происходит увеличение пятого элемента, если 8 — восьмого и т. п.

```
// Анализируемый массив с данными
int Arr[14] = {1, 7, 3, 8, 7, 2, 2, 7, 4, 9, 8, 5, 0, 6};
int Counters[10]; // Вспомогательный массив для подсчета
for (int i=0; i<10; i++) // Очистка вспомогательного массива
    Counters[i] = 0;
for(int i=0; i<14; i++) // Просматриваем все элементы массива
    Counters[Arr[i]]++; // увеличиваем значение соответств.
                        // счетчика
```

Осталось найти индекс максимального значения в массиве Counters. Он и будет модой числа.

После подсчета в Counters будут значения: 1, 1, 2, 1, 1, 1, 1, 3, 2, 1. Больше всего значение в ячейке 7 (т.к. число 7 встречается 3 раза). Мода данного ряда чисел равна 7.

Результат выполнения данного алгоритма позволяет не только находить моду, но и строить гистограмму (график частоты встречаемости) массива данных. Соответствующая функция будет выглядеть так:

```
void ViewGistogram(int *A, int Len)
{
    int max = A[0]; // Для масштабирования находим максимум
    for (int i=1; i<Len; i++) // данных массива A
        if (max < A[i])
            max = A[i];
    for (int i=0; i<Len; i++) // Для каждого значения
    {
        printf("%2d : ", i); // выводим его номер
        int k = (float)A[i] / max * 50; // Максимальная длина – 50
        for (int j=0; j<k; j++) // и строку из * нужного размера
            printf("*");
        printf("\n");
    }
}
```

Вызов функции в вышеописанном примере будет выглядеть так:

```
ViewGistogram(Counters, 10);
```

Медиана ряда чисел – число, стоящее посередине упорядоченного по возрастанию ряда чисел (в случае, если количество чисел нечётное). Если же количество чисел в ряду чётно, то медианой ряда является среднее

арифметическое двух стоящих посередине чисел упорядоченного по возрастанию ряда.

Пример: В ряде чисел 2, 5, 9, 15, 21 медианой является число 9, находящееся посередине.

Пример: Найти медиану чисел 4, 5, 7, 11, 13, 19.

Решение: Здесь четное количество чисел (6). Поэтому ищем не одно, а два числа, записанных посередине. Это числа 7 и 11. Находим среднее арифметическое этих чисел: $(7 + 11) : 2 = 9$.

Число 9 и является медианой данного ряда чисел.

В неупорядоченном ряде чисел:

Для поиска медианы массив предварительно нужно **отсортировать**.

ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

в начале программы ОБЯЗАТЕЛЬНО выводить:
ФИО, группа, номер лаб. работы, номер варианта.

Задание. Написать программу для работы с массивами. Программа должна содержать меню и позволять проводить повторные вычисления. Алгоритмы должны быть реализованы с помощью функций. В программе необходимо задавать размер массивов и выделять под них память динамически. Заполнять массив двумя способами: случайными числами и по порядку от x_1 с шагом k . Выводить массивы и другую информацию на дисплей.

Для получения минимального балла достаточно сделать статические массивы, а в алгоритме вставки дополнительных элементов размер массива можно не увеличивать.

Варианты заданий

1. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, кратные 3. Вывести массив на дисплей. Найти все максимальные элементы и добавить после них 0. Вывести массив на дисплей. Переставить в начало массива все четные числа. Вывести массив на дисплей.

2. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, кратные 5. Вывести массив на дисплей. Найти все минимальные элементы и добавить после них это минимальное значение взятое со знаком минус. Вывести массив на дисплей. Переставить в начало массива все нечетные числа. Вывести массив на дисплей.

3. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые меньше среднего арифметического. Вывести массив на дисплей. Найти все минимальные элементы и добавить после них 0. Вывести массив на дисплей. Переставить в начало массива все числа, кратные 3. Вывести массив на дисплей.

4. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые больше половины максимального значения. Вывести массив на дисплей. Найти все минимальные элементы и добавить после них максимальное значение. Вывести массив на дисплей. Переставить в начало массива все числа, кратные 5. Вывести массив на дисплей.

5. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые больше среднего арифметического. Вывести массив на дисплей. Найти все минимальные элементы и добавить после них -1. Вывести массив на дисплей. Переставить в начало массива все отрицательные четные числа. Вывести массив на дисплей.

6. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые больше K и меньше M (K и M вводятся с клавиатуры). Вывести массив на дисплей. Найти все элементы, кратные 7 и добавить после них 7. Вывести массив на дисплей. Переставить в начало массива все положительные четные числа. Вывести массив на дисплей.

7. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, все элементы, кратные K (K вводится с клавиатуры). Вывести массив на дисплей. Найти все отрицательные элементы и добавить после них максимальный элемент. Вывести массив на дисплей. Переставить в начало массива все положительные числа, кратные 5. Вывести массив на дисплей.

8. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые больше K (K вводится с клавиатуры). Вывести массив на дисплей. Найти все элементы, кратные 3 и добавить после них 3. Вывести массив на дисплей. Переставить в начало массива все положительные нечетные числа. Вывести массив на дисплей.

9. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, кратные 2 и 3. Вывести массив на дисплей. Найти все нечетные элементы и добавить после них это же число, взятое со знаком минус. Вывести массив на дисплей. Переставить в начало массива все отрицательные числа. Вывести массив на дисплей.

10. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые больше среднего арифметического. Вывести массив на дисплей. Найти все отрицательные элементы и добавить после них -100. Вывести массив на дисплей. Переставить в начало массива все числа, кратные 5. Вывести массив на дисплей.

11. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые больше меньше K (K вводится с клавиатуры). Вывести массив на дисплей. Найти все положительные элементы и добавить после них 0. Вывести массив на дисплей. Переставить в начало массива все числа, которые меньше половины максимального значения. Вывести массив на дисплей.

12. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые меньше половины максимального значения. Вывести массив на дисплей. Найти все элементы, которые больше среднего арифметического и добавить после них -1. Вывести массив на дисплей. Переставить в начало массива все отрицательные нечетные числа. Вывести массив на дисплей.

13. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые лежат в диапазоне от K до M (K и M вводятся с клавиатуры). Вывести массив на дисплей. Найти все элементы, кратные 3 и добавить после них 3. Вывести массив на дисплей. Переставить в начало массива все числа, меньшие K. Вывести массив на дисплей.

14. Написать программу для работы с целочисленным массивом. Удалить из массива все отрицательные элементы. Вывести массив на дисплей. Найти все элементы, которые лежат в диапазоне от K до M (K и M вводятся с клавиатуры) и добавить после них число M. Вывести массив на дисплей. Переставить в начало массива все положительные числа, кратные 7. Вывести массив на дисплей.

15. Написать программу для работы с целочисленным массивом. Удалить из массива все положительные нечетные элементы. Вывести массив на дисплей. Найти все элементы, которые меньше K (K вводится с клавиатуры) и добавить после них 0. Вывести массив на дисплей. Переставить в начало массива все нечетные числа. Вывести массив на дисплей.

16. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые больше K (K вводится с клавиатуры). Вывести массив на дисплей. Найти все минимальные элементы и добавить после них число K, взятое со знаком минус. Вывести массив на дисплей. Переставить в начало массива все числа, кратные 3. Вывести массив на дисплей.

17. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые меньше 0 и больше 10. Вывести массив на дисплей. Найти моду этого ряда чисел. Вывести гистограмму на дисплей. Переставить в начало массива все числа, которые меньше 5. Вывести массив на дисплей.

18. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые меньше половины максимального значения. Вывести массив на дисплей. Найти все четные элементы и добавить после них максимальное значение. Вывести массив на дисплей. Переставить в начало массива все числа, кратные 7. Вывести массив на дисплей.

19. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, которые меньше 20 и больше 9. Вывести массив на дисплей. Найти моду этого ряда чисел. Вывести гистограмму на дисплей. Переставить в начало массива все четные числа. Вывести массив на дисплей.

20. Написать программу для работы с целочисленным массивом. Удалить из массива все элементы, кратные 7. Вывести массив на дисплей. Найти все элементы, которые больше K и меньше M (K и M вводятся с клавиатуры) и добавить после них 7. Вывести массив на дисплей. Переставить в начало массива все положительные нечетные числа. Вывести массив на дисплей.