

Массивы.

```
while( n < (docu
i=x
while( n < (docu
{
    n++;
    calC = ev
    i++;
    i++;
```

В рассмотренных ранее примерах алгоритмов производилась обработка одиночных данных — значений простых переменных. При решении практических задач данные объединяются в различные структуры, наиболее простыми из которых являются массивы.

Массив — именованный набор с фиксированным количеством однотипных данных.

В массивы объединены результаты экспериментов, списки фамилий сотрудников, различные сложные структуры данных. Так, список из классного журнала 10 «А» является массивом. В массиве могут быть одинаковые данные, поэтому элементы массива различаются по своим порядковым номерам. Если каждый элемент имеет один порядковый номер, то такой массив называется одномерным, если два — то это таблица из строк и столбцов. Для таблиц первый номер элемента показывает строку, а второй — столбец, на пересечении которых находится элемент. Все строки таблицы имеют одинаковую длину.

Одномерный массив может быть числовой последовательностью с известным количеством членов. Так же, как и в последовательности, в массиве можно указать элемент с конкретным номером, например a_5 , или записать общий вид элемента, используя в качестве индекса переменную и указывая диапазон ее изменения: $a_i, i = 1, 2, \dots, n$.

Задачи на обработку массивов могут иметь различную формулировку. Например, начинаться со слов «Дано n чисел...», а далее говорится, что требуется сделать с этими числами. Чтобы решить такую задачу на компьютере с использованием языка программирования, необходимо выполнить следующее:

- 1) определить, какие числа даны: целые или вещественные (если об этом конкретно не сказано, то лучше считать их вещественными);
- 2) назвать весь массив одним именем, которое будет использоваться для каждого элемента, только к нему добавится номер этого элемента (индекс);
- 3) описать массив в разделе переменных, тем самым отведя место в памяти для массива;
- 4) ввести данные в память.

В описании массива имеется специальное слово **массив**, после которого в квадратных скобках через две точки указывается диапазон изменения номеров элементов, затем слово **из** и пишется тип данных массива. Встретив описание массива, транслятор отводит для него

столько последовательных ячеек, сколько указано в квадратных скобках, и такого формата, каков тип данных массива. Эту память в программе можно использовать целиком или частично, вычисляя значения элементов массива или вводя их с клавиатуры (либо с диска). Чаще всего номера элементов меняются от 1 до заданного числа n . Поместив значение n в разделе констант, в описании можно указать в качестве переменной n последнее значение (верхнюю границу) номера элемента массива.

При обработке массивов решения многих задач основываются на следующих, более простых, задачах: вычисление суммы (произведения) элементов массива; нахождение наибольшего (наименьшего) элемента; упорядочение элементов по возрастанию или убыванию. Рассмотрим эти базовые задачи.

1. Вычисление суммы элементов массива.

Вычисление суммы элементов массива ничем не отличается, в принципе, от суммирования значений простых переменных. Решение задачи состоит из трех основных этапов:

- 1) ввод данных; 2) вычисление суммы; 3) печать результатов.

Выполнение программы вычисления суммы элементов массива представлено в таблице:

Исходные данные: 3, -2, 9, 7, -1, 6, 1							
i	1	2	3	4	5	6	7
$a[i]$	3	-2	9	7	-1	6	1
S	0	1	10	17	16	22	23

2. Нахождение наибольшего элемента массива.

В предыдущем примере производились вычисления, переменная S меняла свои значения в процессе решения задачи. Однако большинство задач, решаемых с помощью компьютера, являются не вычислительными. К ним относится задача поиска наибольшего элемента в массиве. Трудность при разработке алгоритма решения заключается в том, что надо записать в виде команд компьютеру привычные для человека действия: выделение большего из последовательности чисел. Чтобы лучше представить себе, как последовательно просматривать и сравнивать между собой числа, записанные в памяти, вообразим, что каждое число написано на отдельной карточке и карточки сложены стопкой. В таком случае мы первое число запоем сразу как наибольшее и перевернем карточку. Теперь в нашем распоряжении два числа: одно видим, другое — помним. Сравнивая их между собой, запоем большее, т. е. если первое было больше, то запоминать новое не придется и надо смотреть следующую карточку. Если второе больше первого, то первое в

дальнейшем помнить нет смысла и мы запомним второе. Таким образом, на каждом этапе сравнения мы будем помнить большее из просмотренных чисел и в конце решим поставленную задачу. Записав приведенные рассуждения в виде операторов (укрупненных команд), получим программу нахождения наибольшего значения. Промежуточные значения и ответ содержит переменная **max**.

3. Упорядочение массива по возрастанию.

Упорядочения массивов по какому-либо признаку называются также сортировками. Существуют различные методы сортировок, различающиеся, в основном, по скорости получения результата. Рассмотрим один из них — «метод пузырька». Пусть имеется последовательность чисел a_1, a_2, \dots, a_n , которую необходимо упорядочить по возрастанию. Зафиксируем первый элемент и будем последовательно сравнивать его со стоящими справа. Если какой-то из элементов справа окажется меньше первого, то мы поменяем местами этот элемент с первым и продолжим сравнение уже нового элемента, стоящего на первом месте, с оставшимися справа числами. Если снова выявится элемент, меньший зафиксированного, то повторим перестановку. В результате первого просмотра последовательности на первом месте окажется наименьший из всех элементов, т. е. он, как более «легкий», как бы всплывает наверх (отсюда и название метода — «метод пузырька»). Теперь зафиксируем второй элемент и повторим просмотр, выполняя при необходимости перестановки элементов, и т. д. Уяснив идею решения, остановимся на двух вопросах: каким образом фиксировать элементы и как осуществить перестановку двух элементов? Чтобы при переборе элементов, стоящих справа от проверяемого, не менялся индекс последнего, индексы фиксируемого и стоящих правее него элементов должны быть различными: i и j . Индекс i изменяется от 1 до $n - 1$, индекс j всегда на 1 больше i и пробегает все значения от $i + 1$ до n . Для каждого значения i индекс j должен последовательно принять все допустимые значения, следовательно, конструкция программы, отражающая полный перебор всех элементов и их упорядочение по возрастанию, представляет собой двойной цикл. При перестановке двух элементов используется третья переменная. Перестановка местами (обмен значениями в памяти) двух переменных a и b выглядит следующим образом:

1) $c = a$; 2) $a = b$; 3) $b = c$.

4. Поиск элемента в массиве.

Одна из важных не вычислительных задач — поиск данного значения среди элементов массива. Такой поиск называется также

поиском по ключу. На практике поиск осуществляется в упорядоченном массиве, причем имеются различные алгоритмы поиска. Если элемент найден, то напечатаем его номер, если нет, то выдадим соответствующее сообщение. Существенным является то, какой из одинаковых элементов массива, равных данному, нас интересует: первый встретившийся при поиске или последний. Поиск осуществляется в цикле, и как только элемент найден, надо выйти из цикла.

Двумерный массив (или прямоугольная таблица) В из n строк и m столбцов в общем виде выглядит следующим образом:

$$\begin{matrix}
 b_{11} & b_{12} & \dots & b_{1m} \\
 b_{21} & b_{22} & \dots & b_{2m} \\
 \dots & \dots & \dots & \dots \\
 b_{n1} & b_{n2} & \dots & b_{nm}
 \end{matrix}$$

В памяти компьютера элементы двумерного массива расположены один за другим: после элементов первой строки следуют элементы второй строки и т. д. Если число строк таблицы равно числу столбцов, то такая таблица называется квадратной. Главная диагональ квадратной таблицы проходит из левого верхнего угла в правый нижний.

Письменно дополни фразы:

(вставленный текст – подчеркни)

Массив – ...

Например:

Массивы служат для ...

Массивы бывают ...

..... - линейный массив (последовательность

..... - прямоугольная таблица, состоящая из

Каждый элемент массива имеет

Индекс элемента – это

Использование массива начинается с и резервирования

.....

Адреса элементов массива задаются:

- массива
- массива

Метод пузырька

Он заключается в

Поиск по ключу – это