

ЛАБОРАТОРНАЯ РАБОТА №2

МАССИВЫ. УКАЗАТЕЛИ

Цель работы:

Приобрести умения и практические навыки для работы по созданию программ с массивами; приобрести умения и практические навыки по созданию динамических массивов и работе с указателями.

Теоретическая часть:

Массив – конечная поименованная последовательность однотипных величин. Синтаксис объявления массивов выглядит следующим образом:

```
тип имя [размерность] = {инициализатор};
```

Размерность – количество элементов массива. Вместе с типом определяют объем памяти для размещения массива, поэтому размерность может быть указана только константой или константным выражением. Если массив инициализируется при объявлении, то все значения перечисляются через запятую, заключенные в фигурные скобки.

```
int A[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
#define N 100  
float B[N];
```

В памяти элементы массива располагаются последовательно. Если значений инициализировано меньше числа элементов массива, то оставшиеся элементы будут равны нулю. Если значений инициализировано больше – возникнет ошибка. При инициализации массива можно явно не указывать его длину, в этом случае длина массива будет вычисляться исходя из количества его элементов в инициализаторе.

Массив можно объявить константным. В этом случае необходимо его сразу инициализировать, а впоследствии больше не изменять.

К элементам массива обращаются по номерам от 0 до N-1. Массивы обрабатываются поэлементно. Номер элемента массива называется его *индексом*.

Вычислить размер в байтах и количество элементов в массиве можно с

помощью функции *sizeof()*. В данном примере размер массива будет равен 16 байт, так как содержит 4 значения типа *int*:

```
int A[] = { 2, 3, 4, 9};  
size_t size = sizeof(A);  
printf("Размер массива в байтах: %zu", size);
```

Генератор случайных чисел предназначен для получения случайных чисел. Для работы с генератором необходимо присоединить заголовочный файл *<stdlib.h>*.

Функция *rand()* возвращает псевдослучайное целое число в интервале от 0 до значения, присвоенного константе *RAND_MAX*. Значение *RAND_MAX* зависит от системы и определено в заголовочном файле *stdlib.h*. Так, например, оно может быть равно 32767 (двухбайтовое целое) или 2147483647 (четырёхбайтовое целое).

Изменить интервал генерации можно, используя деление по модулю. Например, получить число в интервале от 0 до 99 можно так:

```
x = rand()%100;
```

Получить число в интервале от -100 до 99 можно так:

```
x = rand()%200-100;
```

Функция *srand(int)* устанавливает исходное число для последовательности, генерируемой функцией *rand()*.

Многомерные массивы задаются указанием каждого измерения в квадратных скобках. Двумерные массивы имеют синтаксис:

```
тип имя [размерность][размерность];
```

Количество размерностей массива теоретически не ограничено. В памяти массив располагается построчно. При переходе к следующему элементу памяти быстрее всего меняется самый правый индекс. Многомерный массив можно инициализировать при объявлении. При инициализации можно группировать элементы (построчно или при помощи фигурных скобок). Если элементы сгруппированы, то первую размерность можно не указывать, также можно указывать не все элементы, при этом все

неинициализированные элементы будут равны нулю.

```
#include <stdio.h>

void main() {
    int A[10][10];
    int i, j;
    for(i = 0; i < 10; i++)
        for(j = 0; j < 10; j++)
            scanf_s("%d", &A[i][j]); // ввод элементов
массива с клавиатуры
    for(i = 0; i < 10; i++) {
        printf("\n"); // поэлементный вывод массива
        for(j = 0; j < 10; j++)
            printf("%d \t", A[i][j]);
    }
}
```

Все определенные в программе данные хранятся в памяти по определенному адресу. И указатели позволяют напрямую обращаться к этим адресам и благодаря этому манипулировать данными. *Указатели* представляют собой объекты, значением которых служат адреса других объектов или функций. Указатели — это неотъемлемый компонент для управления памятью в языке Си.

Значением переменной типа указатель является адрес области памяти. Существует несколько видов указателей: на объект, на функции, на void.

Указатель на объект имеет синтаксис *тип *имя;* и содержит адрес значения определенного типа.

```
int *pX;
```

Для вывода значения указателя (адреса объекта) используется специальный спецификатор %p.

Над указателями можно производить различные операции:

- & – получение адреса – применима только к величинам, которые имеют имя и размерность и размещаются в оперативной памяти. Таким

образом, нельзя получить адрес скалярного выражения, неименованной константы;

- * – разыменование (разадресация) – обращение к ячейке памяти по адресу.

Также возможны различные действия над указателями:

- ++, -- – увеличение или уменьшение адреса на длину элемента данного типа;

- увеличение / уменьшение адреса – аналогично предыдущему действию, но вместо единицы адрес увеличивается на *i* длин элементов;

- разность указателей – сколько значений такого типа разделяют два адреса: из первого адреса вычитается второй и делится на длину значения данного типа. Суммирование указателей не определено.

- сравнение – адреса равны, если указывают на один объект.

Указатели на разные типы имеют разные типы соответственно. Эти типы несовместимы.

Указатель на `void` не связан с определенным типом, то есть ему можно присвоить значение любого указателя. Он применяется, когда конкретный тип объекта, адрес которого нужно хранить, неизвестен. Синтаксис следующий: `void *p;`

Данный указатель совместим с указателями любых типов.

Имя массива – адрес его первого элемента, или константный указатель на первый элемент. С массивами можно работать через указатель. Для этого необходимо:

- объявить массив;
- объявить указатель на тип элемента массива;
- присвоить указателю имя массива;
- обратиться к элементам массива через указатель.

```
int A[5] = {1; 2; 3, 4, 5}, B[5];  
int *pA, *pB;
```

```

pA = &A[0];
pB = B;
for (i = 0; i < 5; i++) {
    *pB = *pA;
    pB++;
    pA++;
} // либо *pB++ = *pA++;

```

Существуют ситуации, когда память под переменные нельзя выделить статически. В таких случаях память выделяется динамически (в процессе выполнения программы). Для динамического расширения памяти есть функция, которая содержится в библиотеке *malloc.h*.

Память выделяется при помощи функции *malloc* (*размер требуемого участка памяти*). Она возвращает указатель на `void` – адрес начала выделенной области. Таким образом, обращаться к динамическим переменным приходится через указатель.

```
имя переменной = (тип *) malloc (sizeof (тип));
```

Освобождение выделенного участка памяти производится при помощи команды *free* (*адрес*). Динамические массивы не инициализируются при объявлении, а также не обнуляются. Их преимущество состоит в том, что размерность может определяться в ходе выполнения программы. Доступ к их элементам выполняется так же, как и к элементам статических массивов.

Создание динамического массива:

```

int *m;
int n;
printf("\n Количество элементов: ");
scanf_s("%d", &n);
m = (int *) malloc (n * sizeof (int));

```

Теперь обращаться к элементам данного массива можно также по индексам, например, *m[2]*;

Создание двумерных динамических массивов (из *n* строк и *k* столбцов):

```
int **m, i, j, k, n;
```

```
m = (int **) malloc (n * sizeof (int));  
for (i = 0; i < n; i++)  
    m[i] = (int *) malloc (k * sizeof (int));
```

Задание 1.

Варианты:

1. Заполнить массив $A(10)$ случайными числами. Напечатать количество и сумму отрицательных элементов этого массива.
2. Заполнить массив $A(10)$ случайными числами. Напечатать количество и сумму положительных элементов этого массива.
3. Заполнить массив $M(10)$ случайными числами. Подсчитать и вывести на экран сумму отрицательных элементов массива.
4. Заполнить массив $M(10)$ случайными числами. Подсчитать и вывести на экран сумму положительных элементов массива.
5. Заполнить массив $mas(10)$ случайными числами. Подсчитать и вывести на экран количество четных элементов массива.
6. Заполнить массив $mas(10)$ случайными числами. Подсчитать и вывести на экран количество нечетных элементов массива.
7. Заполнить массив $A(10)$ случайными числами. Подсчитать и вывести на экран количество элементов массива, кратных числу, введенному с клавиатуры.
8. Заполнить массив $X(10)$ случайными числами. Подсчитать и вывести на экран количество элементов массива, значения которых лежат в интервале от 25 до 50.
9. Заполнить массивы $X(15)$ и $Y(15)$, случайными числами. Вывести все числа массива Y , которые не входят в массив X .
10. Заполнить массивы $X(15)$ и $Y(15)$, случайными числами. Вывести все числа массива Y , которые входят в массив X .
11. Заполнить массив $A(15)$ случайными числами от 1 до 50. Найти и вывести сумму элементов с нечетными индексами.
12. Заполнить массив $A(15)$ случайными числами от 1 до 50. Найти и вывести сумму элементов с четными индексами.
13. Дан массив $A(12)$. Найти в нем максимальный и минимальный элементы, вывести эти элементы и их индексы.

14. Заполнить массив A (20) случайными числами. Найти и вывести среднее арифметическое четных элементов массива.
15. Заполнить массив A (20) случайными числами. Найти и вывести среднее арифметическое нечетных элементов массива.
16. Дан массив $A(10)$, заполненный случайными числами. Определить количество элементов, меньших среднего арифметического массива.
17. Дан массив $A(10)$, заполненный случайными числами. Определить количество элементов, больших среднего арифметического массива.
18. Заполнить массив $A(30)$ случайными числами. Определить, есть ли в массиве повторяющиеся элементы.
19. Дан массив $A(15)$, заполненный случайными числами. Поменять местами первый и последний элементы.
20. Заполнить массив $A(10)$ случайными числами от 1 до 100. Найти второй по величине элемент массива.
21. Дан массив $A(12)$. Перенести все нулевые элементы в конец массива, сохранив порядок остальных элементов.
22. Дан массив $A(10)$, заполненный случайными числами. Проверить, является ли массив палиндромом.
23. Дан массив $A(9)$, заполненный случайными числами. Заменить все отрицательные элементы их модулями.
24. Дан массив $A(10)$, заполненный случайными числами. Заменить все положительные элементы нулями.
25. Дан массив $A(16)$, заполненный случайными числами. Вывести его элементы в обратном порядке.
26. Дан массив $A(10)$, заполненный случайными числами. Подсчитать, сколько раз в нем встречается число, введенное с клавиатуры.
27. Заполнить массив $A(10)$ случайными числами и циклически сдвинуть его элементы вправо на 3 позиции.

Задание 2.

1. Дан массив $F(5, 5)$, заполненный случайными числами. Вывести его на экран. Вывести на экран индексы элементов, значение которых лежит в интервале от 10 до 30.

2. Дан массив $C(6,6)$ и массив $B(6,6)$, заполненные случайными числами. Образовать новый массив A такой, что $A_{ij} = \min(B_{ij}, C_{ij})$. Вывести на экран полученный массив.

3. Дан массив $C(6,6)$ и массив $B(6,6)$, заполненные случайными числами. Образовать новый массив A такой, что $A_{ij} = \max(B_{ij}, C_{ij})$. Вывести на экран полученный массив.

4. Дан массив $X(10,10)$, заполненный случайными числами. Вывести его на экран. Найти и вывести на экран максимальный элемент массива.

5. Дан массив $X(10,10)$, заполненный случайными числами. Вывести его на экран. Найти и вывести на экран минимальный элемент массива.

6. Дан массив $A(10 \times 10)$, заполненный случайными числами. Найти сумму отрицательных элементов из закрашенной на рисунке области.



7. Дан массив $A(10 \times 10)$, заполненный случайными числами. Найти сумму отрицательных элементов из не закрашенной на рисунке области.



8. Дан массив $A(10 \times 10)$, заполненный случайными числами. Найти сумму положительных элементов из закрашенной на рисунке области.



9. Дан массив $A(10 \times 10)$, заполненный случайными числами. Найти сумму положительных элементов из не закрашенной на рисунке области.



10. Даны массивы $A(5,5)$ и $B(5,5)$, заполненные случайными числами. Вывести на экран все числа, которые находятся одновременно в этих двух массивах.

11. Даны массивы $A(5,5)$ и $B(5,5)$, заполненные случайными числами. Вывести на экран все числа, которые различны в этих двух массивах.

12. Дан двумерный массив. Преобразовать его в одномерный массив, читая по спирали. Вывести на экран полученный одномерный массив.

13. Заполнить массив $A(5,5)$ случайными числами. Найти сумму элементов главной диагонали.

14. Дан массив $A(3,3)$. Поменять местами первую и последнюю строки.

15. Заполнить массив $A(7,7)$ числами от 1 до 49. Определить сумму элементов в последнем столбце.

16. Дан массив $A(5,5)$. Найти наибольший элемент во всей матрице и его индексы.

17. Дан массив $A(5,5)$. Отсортировать каждую строку по возрастанию.

18. Заполнить массив $A(6,6)$ случайными числами. Найти строку с наибольшей суммой элементов.

19. Дан массив $A(4,4)$. Найти сумму элементов каждого столбца и вывести их.

20. Заполнить массив $A(6,6)$ случайными числами. Поменять местами строки с максимальной и минимальной суммой элементов.

21. Заполнить массив $A(6,6)$ случайными числами. Поменять местами столбцы с максимальной и минимальной суммой элементов.

22. Дан массив $A(5,5)$. Вычислить сумму элементов побочной диагонали.
23. Заполнить массив $A(6,6)$ случайными числами. Определить количество четных и нечетных элементов.
24. Заполнить массив $A(5,5)$ случайными числами. Найти среднее арифметическое всех элементов массива.
25. Дан массив $A(4,4)$. Заменить все элементы, кратные 3, на их квадраты.
26. Дан массив $A(8,8)$. Заменить все четные элементы нулями.

**Задание 3. Числа M и N в данном задании вводить с клавиатуры.
Диапазон генерации чисел также принимать с клавиатуры.**

1. Даны два массива: $A[N]$ и $B[M]$. Заполнить массивы случайными числами. Пользуясь указателями создать третий массив, в котором нужно собрать элементы обоих массивов.
2. Даны два массива: $A[N]$ и $B[M]$. Заполнить массивы случайными числами. Пользуясь указателями создать третий массив, в котором нужно собрать элементы массива A , которые не включаются в B .
3. Даны два массива: $A[N]$ и $B[M]$. Заполнить массивы случайными числами. Пользуясь указателями создать третий массив, в котором нужно собрать элементы массива B , которые не включаются в A .
4. Даны два массива: $A[N]$ и $B[M]$. Заполнить массивы случайными числами. Пользуясь указателями создать третий массив, в котором нужно собрать элементы массивов A и B , которые не являются общими для них.
5. Даны два массива: $A[N]$ и $B[M]$. Заполнить массивы случайными числами. Пользуясь указателями создать третий массив, в котором нужно собрать элементы массивов A и B , которые являются общими для них.
6. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Удвоить значения элементов массива, которые стоят на четных местах.
7. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Удвоить значения элементов массива, которые стоят на нечетных местах.
8. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Удвоить значения элементов массива, которые стоят на местах, кратных вводимому с клавиатуры числу.
9. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Найти в нем максимальный и минимальный элементы и вывести их на экран.
10. Создать одномерный динамический массив из N целых чисел.

Заполнить массив случайными числами. Заменить отрицательные числа их модулями.

11. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Заменить четные числа числами в два раза меньше.

12. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Заменить числа, имеющие натуральный корень, этим корнем.

13. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Найти среднее арифметическое всех элементов и заменить все элементы, меньшие среднего, на само среднее значение.

14. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Перевернуть массив (развернуть его задом наперед) без использования дополнительного массива.

15. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Определить количество уникальных значений в массиве.

16. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Определить количество значений в массиве, встречающихся более одного раза.

17. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Найти и вывести наиболее часто встречающийся элемент.

18. Создать одномерный динамический массив из N целых чисел. Заполнить массив случайными числами. Поменять местами первый положительный и первый отрицательный элементы.

Вопросы к теоретическому материалу

1. Что называется массивом?
2. Укажите синтаксис объявления массивов.
3. Что называется размерностью массива?
4. Какая функция используется для генерации случайных чисел?
5. Как сгенерировать число в каком-либо промежутке?
6. Какой заголовочный файл хранит функцию генерации случайных чисел?
7. Какая функция устанавливает исходное число для генератора случайных чисел?
8. Укажите синтаксис объявления двумерного массива.
9. Ограничено ли количество размерностей массива?
10. Как найти количество элементов массива?
11. Что называется указателем?
12. Перечислите операции, которые можно производить над указателями.
13. Охарактеризуйте операцию получения адреса, производимую над указателями.
14. Охарактеризуйте операцию разыменования, производимую над указателями.
15. Перечислите действия, которые можно производить над указателями.
16. Охарактеризуйте нахождение разности указателей. Укажите, для чего это можно применять. Определено ли суммирование указателей?
17. Охарактеризуйте указатель на void.
18. Опишите последовательность действий при работе с массивами через указатель.
19. Какая функция применяется для динамического выделения памяти? В какой библиотеке она содержится?
20. Что принимает функция для динамического выделения памяти?

21. Что возвращает функция для динамического выделения памяти?
22. Укажите, каким образом выделяется память под одну переменную при помощи функции `malloc`.
23. Укажите, при помощи какой команды происходит освобождение выделенного участка памяти.
24. Кратко опишите процесс создания динамического массива.

ПРОЦЕСС СДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

По итогам выполнения каждой лабораторной работы студент:

1. Демонстрирует преподавателю правильно работающие программы;
2. Демонстрирует приобретенные теоретические знания, отвечая на пять вопросов по лабораторной работе;
3. Демонстрирует отчет по выполненной лабораторной работе, соответствующий всем требованиям.

Отчет по лабораторной работе оформляется по шаблону, представленному в приложении 1. Требования к отчету представлены в приложении 2.

ПРИЛОЖЕНИЕ 1. ШАБЛОН ОТЧЕТА
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Казанский национальный исследовательский технический университет им.
А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации
Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА №
по дисциплине
СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Работу выполнил

Студент гр.43__

Фамилия И.О.

Принял

Преподаватель Григорьева В.В.

Казань, 2025 г.

1. **Цель работы.**

2. **Задание на лабораторную работу** – вставляется задание на лабораторную работу, соответствующее индивидуальному, выданному преподавателем, варианту студента.

3. **Результат выполнения работы** – формируется описание хода выполнения работы и вставляются скриншоты с результатами работы разработанных программ (скриншоты должны быть подписаны, например, *Рисунок 1. Начальное состояние программы* и т.п.).

4. **Листинг программы** – вставляется код разработанной программы

ПРИЛОЖЕНИЕ 2. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА

Лист документа должен иметь книжную ориентацию, поля документа должны составлять: левое – 3 см, правое – 1,5 см, верхнее – 2 см, нижнее 2 см.

Нумерация страниц – внизу страницы по центру, первая страница не нумеруется

Междустрочный интервал – 1,5 (полуторный), отступ первой строки – 1,25.

Текст документа должен быть выполнен с использованием шрифта Times New Roman, размер – 14, выравнивание – по ширине. Заголовки выполняются тем же шрифтом, но размера 16, полужирное начертание, размещение – по центру.

Рисунки должны размещаться по центру, они нумеруются по порядку. Перед рисунком в тексте на него должна быть ссылка. Подпись рисунка должна располагаться по центру и быть выполнена шрифтом Times New Roman, размер – 12. Сначала происходит нумерация рисунка, а затем пишется его название.