

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

Результаты работы, фотоотчет, предоставить преподавателю на e-mail: r.bigangel@gmail.com **до 17.04.2023.**

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 № 15

Тема: «Понятие многомерного массива. Обработка элементов матриц»

Цель: изучить понятие матрицы, селективной обработки элементов матрицы, а также обработки элементов столбцов, строк и диагоналей матриц.

Теоретические сведения

Двухмерные массивы, матрицы

Двухмерный массив представляет собой список одномерных массивов.

Общая форма записи двухмерного массива:

тип имя_массива[размер1] [размер2];

В приведенной записи размер1 означает количество строк двухмерного массива, а размер2 – количество столбцов.

В двухмерном массиве позиция любого элемента определяется двумя индексами. Индексы каждого из размеров массива начинаются с 0 (с нуля). Место хранения для всех элементов массива определяется во время

компиляции. Память, выделенная для хранения массива, используется в течение всего времени существования массива.

Для двумерных массивов общий размер массива в байтах вычисляется по формуле

всего байт = число строк * число столбцов * размер типа в байтах

Многомерные массивы

Общая форма записи многомерного массива:

тип имя_массива[размер1] [размер2]... [размерN];

Индексация каждого размера начинается с нуля. Элементы многомерного массива располагаются в памяти в порядке возрастания самого правого индекса. Поэтому правый индекс будет изменяться быстрее, чем левый (левые).

При обращении к многомерным массивам компьютер много времени затрачивает на вычисление адреса, так как при этом приходится учитывать значение каждого индекса. Следовательно, доступ к элементам многомерного массива происходит значительно медленнее, чем к элементам одномерного. В этой связи использование многомерных массивов встречается значительно реже, чем одномерных или двумерных массивов.

Для многомерных массивов общий размер многомерного массива в байтах вычисляется по формуле

*всего байт = размер1 * размер2 * ... * размерN * размер типа в байтах*

Очевидно, многомерные массивы способны занять большой объем памяти, а программа, которая их использует, может очень быстро столкнуться с проблемой нехватки памяти.

Для определения размера типа в байтах применяется функция `sizeof()`, которая возвращает целое число. Например, `sizeof(float)`.

Инициализация массивов

При инициализации многомерного массива для улучшения наглядности элементы инициализации каждого измерения можно заключать в фигурные скобки.

Пример инициализации двумерного массива:

```
int    MN[3][4] = {  
    {1, 2, 3, 4},  
    {5, 6, 7, 8},  
    {9, 10, 11, 12}  
};
```

Массив MN[3][4] – это матрица, у которой 3 строки и 4 столбца.

Для многомерных массивов инициализацию можно также проводить с указанием номера инициализируемого элемента.

Пример инициализации трехмерного массива:

```
int    XYZ[2][3][4] = {  
    {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}},  
    {{13, 14, 15, 16}, {17, 18, 19, 20}, {21, 22, 23, 24}}  
};
```

Как видно, массив XYZ содержит два блока, каждый из которых есть матрица размера 3×4 , т. е. 3 строки и 4 столбца.

В многомерном массиве размер самого левого измерения также можно не указывать. В частности, для инициализации массива MN[3][4] допустима следующая запись:

```
int    MN[][4] = {  
    {1, 2, 3, 4},  
    {5, 6, 7, 8},  
    {9, 10, 11, 12}  
};
```

При инициализации многомерных массивов необходимо указать все данные (размерности) за исключением крайней слева размерности. Это нужно для того, чтобы компилятор смог определить длину подмассивов, составляющих массив, и смог выделить необходимую память. Рассмотрим пример безразмерной инициализации для трехмерного массива целых чисел:

```
int    XYZ[][3][4] = {
```

```

{
{1, 2, 3, 4},
{5, 6, 7, 8},
{9, 10, 11, 12}
},
{
{13, 14, 15, 16},
{17, 18, 19, 20},
{21, 22, 23, 24}
}
};

```

Вывод трехмерного массива на консоль (дисплей) можно выполнить по следующей программе:

```

#include <stdio.h>
#include <conio.h>
int main (void) {
int i, j, k;
int XYZ[][3][4] = {
{ {1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12} }, // 1-й
{ {13, 14, 15, 16}, {17, 18, 19, 20}, {21, 22, 23, 24} } }; //
2-й
for (i = 0; i < 2; ++i) { printf("\n");
for (j = 0; j < 3; ++j) { printf("\n"); for (k = 0; k < 4; ++k)
printf(" %3d", XYZ[i][j][k]);
}
}
printf("\n\n Press any key: ");
_getch(); return 0;
}

```

Практическая часть

Пример 1. Напишите программу заполнения квадратной матрицы (заданного размера $n > 2$) по спирали натуральными числами начиная с левого верхнего угла (принимая его за номер 1) и двигаясь по часовой стрелке.

Образец заполнения:

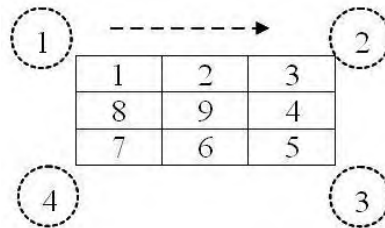


Рисунок 6.1 – Образец заполнения матрицы числами по спирали

Программный код решения примера:

```
#include <stdio.h>
#include <conio.h>
#define n 13

int main(void) {
    int i = 1, j, k; int p = n/2; int A[n][n];
    // Обнуление матрицы for (j = 0; j < n; ++j)
    for (k = 0; k < n; ++k) A[j][k] = 0;
    printf("\n\t Spiral matrix of dimention (%d x %d):\n", n, n); for (k = 1; k <=
p; k++) // Число спиралей
    {
        // Верхний горизонтальный столбец for (j = (k-1); j < (n-k+1); j++)
        A[(k-1)][j] = i++;
        // Правый верхний столбец
        for (j = k; j < (n-k+1); j++)
            A[j][n-k] = i++;
        // Нижний горизонтальный столбец for (j = (n-k-1); j >= (k-1); --j)
```

```

A[n-k][j] = i++;
// Левый верхний столбец
for (j = (n-k-1); j >= k; j--)
A[j][(k-1)] = i++;
}
if ( n % 2 )
A[p][p] = n*n;
// Распечатка матрицы for (i = 0; i < n; ++i)
for (j = 0; j < n; ++j)
{
printf("%5d", A[i][j]); if (j == (n-1))

}

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

```

Результат выполнения программы показан на рис. 6.2.

```

C:\Users\USER\documents\visual studio 2010\Projects\lab_6\Debug\lab_6.exe
Spiral matrix of dimention <13 x 13>:
 1  2  3  4  5  6  7  8  9  10  11  12  13
48 49 50 51 52 53 54 55 56 57 58 59 60
47 88 89 90 91 92 93 94 95 96 97 60 15
46 87 120 121 122 123 124 125 126 127 98 61 16
45 86 119 144 145 146 147 148 149 128 99 62 17
44 85 118 143 160 161 162 163 150 129 100 63 18
43 84 117 142 159 168 169 164 151 130 101 64 19
42 83 116 141 158 167 166 165 152 131 102 65 20
41 82 115 140 157 156 155 154 153 132 103 66 21
40 81 114 139 138 137 136 135 134 133 104 67 22
39 80 113 112 111 110 109 108 107 106 105 68 23
38 79 78 77 76 75 74 73 72 71 70 69 24
37 36 35 34 33 32 31 30 29 28 27 26 25
Press any key:

```

Рисунок 6.2 – Заполнение матрицы по спирали

Пример 2. Каждый день производятся замеры некоторых величин (вещественных значений), причем значения этих величин сводятся в прямоугольную таблицу размера $n \times m$. Составьте многомерный массив данных за 30 дней. Формирование данных произвести по случайному равномерному закону из интервала от -12 до 21 .

Этот пример относится к определению трехмерного массива данных.

Программный код решения примера:

```
#include <stdio.h>
#include <conio.h>
#include <time.h>
#include <stdlib.h>
#define n 6
#define m 7
#define N 30
const int Left = -12; // Левая граница const int Right = 21; // Правая
граница

int main (void)
{
float R, r;
float A[N][n][m]; int i, j, k;
// Инициализация генератора случайных чисел srand((unsigned)
time(NULL));
printf("\n\t The values of every 10 days from 30 days:");// Формирование
данных за 30 дней
for (k = 0; k < N; ++k)
for (i = 0; i < n; ++i)
for (j = 0; j < m; ++j)
{
r = (float) rand()/RAND_MAX; R = Left + (Right - Left)*r;
```

```

A[k][i][j] = R;}
// Печать данных за каждый 10-й день
for (k = 0; k < N; k += 10) { printf("\n");
for (i = 0; i < n; ++i) { printf("\n"); for (j = 0; j < m; ++j)
printf("%10.4f", A[k][i][j]);
}}
printf("\n Press any key: ");
_getch(); return 0;
}

```

В программе используется трехмерный массив размера $30 \times 6 \times 7$. Это означает, что прямоугольная таблица (массив) данных размера 6×7 как бы скрепляется 30 раз – по заданному числу дней. Границы случайных чисел определены с помощью спецификатора **const**.

Возможный результат выполнения программы показан на рис. 6.3.

```

C:\Users\USER\documents\visual studio 2010\Projects\lab_6\Debug\lab_6.exe
The values of every 10 days from 30 days:
11.4063  12.8354  18.4218  -10.8912  14.0096  17.9877  18.7954
 7.3315  -2.2038  -8.1891   2.1519  14.6482  -0.9178   1.0723
16.5476  -3.6722  20.2447  15.5817   7.2227   8.1221   5.1038
19.1701  -0.6942  -4.7065   6.4412   6.2609  -3.2885   1.0209
10.4354  17.6977  -5.6945   2.0583  -11.7120 -10.0331 -11.1762
12.0780   2.8388   0.5043  -10.3856  20.5831  16.4307  -9.7934

 8.0949   0.1971  -0.1503  -7.1195  -2.2894   3.2678  -5.4961
 7.4342  -5.8496  17.7289  -3.4365  -8.4177  20.7734  18.5084
-4.2714  18.2647  13.1516  16.2424  -3.4819  14.1265  -8.7712
 6.9297  -8.0562   7.6296  -4.3510   2.0784  -7.3230  12.6641
-4.4789  12.5201  -2.7537  -9.9113  -8.5758   7.5279   3.2869
16.1115  -7.0843  19.4219  11.3791  10.8856  -4.6682  19.2809

 4.6677  20.2185 -10.4783   0.7933  -5.2201  -4.1466  -7.2746
 5.7533  -6.5062  18.0421  -9.1338  19.1459  17.0773 -10.6092
 9.3638 -10.5216  -2.7900   7.7605  -6.5455   8.0979  -5.8073
-5.2594  19.3806   9.7526  -6.1346  12.0417  -7.1608 -11.6536
 6.9911   0.4439  -9.2365  -1.2380  18.4117   2.4762   9.9188
-8.8336   9.7365  -0.5391  14.3752   2.7209  -1.1433  -4.8999
Press any key:

```

Рисунок 6.3 – Вывод данных за каждый 10-й день

Содержание задания:

1. В соответствии с вариантом составить блок-схемы и программы для решения следующих задач таблица 1 (задача **а** на селективную обработку элементов матрицы, задача **б** на обработку элементов строк и столбцов).

2. Отсортировать полученный массив тремя способами

- Селективная обработка элементов массива
- Сортировка выбором
- Сортировка методом пузырька

3. Подготовить отчет по лабораторной работе.

Вариант	Содержание задачи	
1	а)	Найти сумму положительных кратных 3 элементов.
	б)	Найти среднее геометрическое нечетных элементов 2-го столбца и количество кратных 5 элементов 3-ей строки матрицы D(5;5).
2	а)	Найти количество отрицательных четных элементов.
	б)	Найти сумму нечетных элементов 3-го столбца и произведение отрицательных кратных 3 элементов 2-ой строки матрицы C(6;6).
3	а)	Найти произведение положительных четных элементов.
	б)	Найти произведение суммы кратных 3 чисел в 4-ом столбце на количество нечетных чисел 2-ой строки матрицы T(4;6).
4	а)	Найти сумму отрицательных четных элементов.
	б)	В матрице A(5;7) найти разность количества нечетных чисел 1-ой строки и количества четных чисел 4-го столбца.
5	а)	Найти максимальный элемент, номер строки и столбца, в котором он находится.
	б)	Найти сумму нечетных элементов 2-го столбца и произведение отрицательных кратных 3 элементов 4-ой строки матрицы D(4;4).
6	а)	Найти минимальный элемент, номер строки и столбца, в котором он находится.
	б)	Найти произведение отрицательных четных элементов 2-ой строки и количество не кратных 5 элементов 2-го столбца матрицы B(3;5).
7	а)	Найти количество положительных кратных 5 элементов.
	б)	Подсчитать количество положительных кратных 3 элементов 1-ой строки и количество нечетных элементов 2-го столбца матрицы A(6;6).
8	а)	Найти произведение отрицательных нечетных элементов.
	б)	Найти разность произведения нечетных чисел 3-ей строки и произведения отрицательных чисел 1-го столбца матрицы B(4;4).
9	а)	Найти квадрат минимального элемента и номер строки и столбца, где он находится.
	б)	В матрице T(3;9) найти разность произведения нечетных чисел 2-ой строки и суммы положительных чисел 6-го столбца.

10	a)	Найти произведение положительных не кратных 5 элементов.
	б)	В матрице $A(5;5)$ найти сумму количества четных чисел 3-ей строки и количества отрицательных чисел 4-го столбца.
11	a)	Найти максимальный по модулю элемент и номер строки и столбца, где он находится.
	б)	В матрице $C(5;6)$ найти произведение количества нечетных чисел 2-го столбца и количества положительных чисел 3 строки.
12	a)	Найти количество отрицательных не кратных 3 элементов.
	б)	Найти максимальный элемент 2-ой строки и количество четных элементов 5-го столбца матрицы $X(5;5)$.
13	a)	Найти произведение положительных нечетных элементов.
	б)	В матрице $A(6;6)$ найти произведение суммы четных чисел 3-ей строки и суммы отрицательных чисел 1-го столбца.
14	a)	Найти сумму отрицательных нечетных элементов.
	б)	Найти произведение суммы положительных чисел 1-й строки на сумму четных чисел 2-го столбца матрицы $M(4;5)$.
15	a)	Найти произведение отрицательных четных элементов.
	б)	В матрице $A(7;7)$ найти разность количества отрицательных чисел 2-ой строки и количества нечетных чисел 3-го столбца.

Контрольные вопросы:

- 1 Как организуются многомерные числовые массивы в языке C++?
- 2 Как организуется индексирование числовых массивов в языке?
- 3 На кого или на что возлагается контроль границ числовых массивов в языке программирования C++?
- 4 В какой очередности и как происходит заполнение многомерных числовых массивов в программах на языке C++?
- 5 Для чего применяется начальная инициализация числовых массивов при дальнейшем их использовании?
- 6 Сколько потребуется операторов цикла для вывода на консоль двухмерного числового массива (матрицы чисел)?
- 7 Почему при определении размерности массива с помощью препроцессорной директивы **define** не используется точка с запятой после числового значения?