

ЛАБОРАТОРНАЯ РАБОТА № 14

МНОГОМЕРНЫЕ МАССИВЫ

Цель работы: получение практических навыков в работе с двумерными массивами.

Краткие теоретические сведения

Массивы могут быть многомерными. Самые распространенные многомерные массивы – это матрицы (двумерные массивы). Объявляются они следующим образом:

тип_элементов имя [m][n];

где m – это количество строк, n – количество столбцов.

Нумерация строк и столбцов начинается с нуля.

Например, объявление

float b[4][6];

означает массив из четырех элементов, каждый из которых есть массив из шести вещественных чисел. Отдельные величины этого массива обозначаются именами с двумя индексами: b[0][0], b[0][1], ..., b[3][5].

Порядок расположения элементов многомерного массива в памяти такой, что прежде всего меняется последний индекс, затем предпоследний и т. д., и лишь один раз пробегает свои значения первый индекс.

Существуют два варианта объявления массива:

- неинициализированный (пример выше);
- инициализированный. Определяется следующим образом: int b[3][4]={ { 2, 3, 5, 7 },
 { 0, 8, 12, 1 },
 { 1, -8, 6, 9 } };

Если в строке указано меньше элементов, чем требуется, то остальные инициализируются нулями.

При работе с матрицами организуются два цикла. Внешний цикл отвечает за перебор всех строк матрицы, а внутренний перебирает все элементы строки.

Ввод элементов матрицы

```
const int m=3, n=5; // количество строк и столбцов
float b[m][n]; // объявление массива
int i, j; // объявление индексов элементов
for (i=0; i<m; i++)
{ puts("Введите %d-ю строку", i); // печать подсказки
  for (j=0; j<n; j++)
    scanf("%f", &b[i][j]); // считывание с клавиатуры элемента
}
```

Функция `scanf` обеспечивает каждый раз ввод одного элемента массива, т. е. после набора на клавиатуре значения очередного элемента нужно нажимать пробел, а после последнего элемента строки – клавишу Enter.

Ввод по строкам является наиболее естественным. В случае необходимости можно организовать заполнение массива по столбцам. Для этого нужно поменять местами операторы `for`:

```
const int m=3, n=5; // количество строк и столбцов
float b[m][n]; // объявление массива
int i, j; // объявление индексов элементов
for (j=0; j<n; j++)
{ puts("Введите %d-й столбец", j); // печать подсказки
  for (i=0; i<m; i++)
    scanf("%f", &b[i][j]); // считывание с клавиатуры элемента
}
```

Заполнение матрицы случайными числами

Выполняется также в двойном цикле аналогично одномерным массивам. В примере показано заполнение целой матрицы случайными числами в интервале **[a,b]** (для вещественных чисел формула изменится – см. одномерные массивы). В этой и последующих программах будем считать, что объявлена целая матрица **М** на **N**, где **М** и **N** — целые константы (объявленные через *const*), а также целые переменные *i* и *j*.

```
for ( i = 0; i < M; i ++ )
    for ( j = 0; j < N; j ++ )
        A[i][j] = random(b-a+1) + a;
```

Вывод элементов матрицы

Двумерные массивы необходимо выводить в привычном виде (по строкам), начиная вывод новой строки массива в новую строку экрана.

```
for ( i = 0; i < M; i ++ )
    { for ( j = 0; j < N; j ++ )
        printf(“%d \t”, A[i][j]);
      printf (“\n”);
```

Суммирование элементов матрицы

```
for (S=0, i=0; i<m; i++)
    for (j=0; j<n; j++)
        S=S+b[i][j];
```

Суммирование диагональных элементов матрицы (вычисление следа матрицы)

```
for (S=0, i=0; i<n; i++)
    S=S+b[i][i];
```

Суммирование элементов заданной строки матрицы

Для двумерного массива **b** размером **M** на **N** необходимо вычислить сумму элементов строки с номером **k**.

```
for (S=0, j=0; j<n; j++)
    S=S+b[k][j];
```

Суммирование элементов строк матрицы

Необходимо вычислить сумму элементов каждой строки двумерного массива **b** размером **M** на **N** с выводом результатов на экран.

```
for (i=0; i<m; i++)
    { S=0;    // обнуление суммы каждой строки
      for (j=0; j<n; j++)
```

```

        S=S+b[i][j];
    printf ("Сумма элементов строки %d=%d \n ", i, S); }

```

Транспонирование матрицы

Необходимо заменить строки матрицы ее столбцами, а столбцы – строками.

```

int b[m][n], a[n][m], i, j;
....
for (i=0; i<n; i++)
    for (j=0; j<m; j++)
        a[i][j]=b[j][i];

```

Для квадратной матрицы размером N на N для этого необходимо поменять местами каждый элемент верхнего треугольника с соответствующим элементом нижнего (диагональные элементы переставлять не нужно). При этом для каждой строки нужно выполнять перестановку элементов, расположенных правее главной диагонали, с элементами соответствующего столбца, расположенными ниже главной диагонали. При перестановке используем вспомогательную переменную P , помещая в нее для временного хранения один из переставляемых элементов, чтобы не потерять его значение.

```

int b[n][n], i, j, P;
....
for (i=0; i<n-1; i++)
    for (j=i+1; j<n; j++)
        { P= b[i][j];
          b[i][j]=b[j][i];
          b[j][i]=P;
        }

```

Умножение матрицы на вектор

```

int a[m][n], i, j, s, b[n], c[m];
....
for (i=0; i<m; i++)
    { s=0;
      for (j=0; j<n; j++)

```

```

        s=s+a[i][j]*b[j];
    c[i]=s;
}

```

Удаление строки из матрицы

Требуется удалить строку с заданным номером k. Решение задачи аналогично удалению элемента из одномерного массива. Все строки, начиная с (k+1)-й, нужно переместить вверх. Число строк уменьшается на 1.

```

for (i=k; i<m; i++)
    for (j=0; j<n; j++)
        b[i][j]=b[i+1][j];

```

Удаление столбца осуществляется аналогично.

Перестановка строк матрицы

1-й способ.

С использованием вспомогательной переменной P (того же типа, что и элементы массива) перестановка осуществляется во всех столбцах двух строк.

```

for ( k=0; k<n; k++)
    { P=A[i][k]; A[i][k]=A[j][k]; A[j][k]=P; }

```

2-й способ

С использованием вспомогательного массива, в который целиком пересылается одна из строк для временного хранения.

```

for ( k=0; k<n; k++)
    c[k]=a[i][k];
for ( k=0; k<n; k++)
    a[i][k]= a[j][k];
for ( k=0; k<n; k++)
    a[j][k]=c[k];

```

Примеры решения

1. Вычисление и вывод на экран таблицы умножения в форме матрицы Пифагора.

```

#include <stdio.h>
#include<conio.h>
void main()
{ const int n=10;
  int a[n][n], i, j;
  clrscr();
  for (i=1; i<n; i++)
  { for ( j=1; j<n; j++)
    { a[i][j]=i*j;
      printf ("%5d", a[i][j] );
    }
    printf ("\n");
  }
  getch();
}

```

По данной программе в двумерном массиве не будут заполнены нулевая строка и нулевой столбец.

2. Заполнение матрицы случайными числами в диапазоне от 0 до 99 и поиск в ней максимального значения.

```

#include <stdio.h>
#include<conio.h>
#include <stdlib.h>
#define n 5
void main()
{
  int a[n][n], i, j, imax, jmax;
  clrscr();
  randomize();
  for (i=0; i<n; i++)
  { for ( j=0; j<n; j++)
    { a[i][j]=rand() % 100;
      printf ("%d\t", a[i][j]);
    }
    printf ("\n");
  }
  imax=jmax=0;           // начало поиска
}

```

```

for (i=0; i<n; i++)
    for ( j=0; j<n; j++)
        if (a[i][j]>a[imax][jmax])
            { imax=i; jmax=j; }
printf ("max=a[%d][%d]=%d\n", imax, jmax, a[imax][jmax]);
getch();
}

```

3. Дан двумерный массив различных вещественных чисел, содержащий 5 строк и 4 столбца. Строку, содержащую максимальный элемент массива, поменять местами со строкой, содержащей минимальный элемент:

```

#include <stdio.h>
void main()
{ float m1[5][4], max, min, m2[4];
  int maxi, mini, i, j;
  puts("Введите исходный массив: \n");
  for(i=0;i<5;i++)
      for(j=0;j<4;j++)
          scanf("%f", &m1[i][j]);
  max=m1[0][0];
  min=m1[0][0];
  for(i=0;i<5;i++)
      for(j=0;j<4;j++)
          if(max<=m1[i][j])
              { max=m1[i][j];
                maxi=i;}
  for(i=0;i<5;i++)
      for(j=0;j<4;j++)
          if(min>=m1[i][j])
              { min=m1[i][j];
                mini=i;}
  for(j=0;j<4;j++)
      m2[j]=m1[maxi][j];
  for(j=0;j<4;j++)
      m1[maxi][j]=m1[mini][j];
  for(j=0;j<4;j++)
      m1[mini][j]=m2[j];
}

```

```
puts("Массив после перестановки строк:\n");
for(i=0;i<5;i++)
{for(j=0;j<4;j++)
printf("%f ", m1[i][j]);
printf("\n");}
}
```

Порядок выполнения работы

1. Изучить теоретические сведения.
2. Ответить на контрольные вопросы.
3. Выполнить задание.

Контрольные вопросы

1. Что такое двумерный массив?
2. Сколько индексов требуется для обращения к элементу двумерного массива и что определяет каждый из индексов?
3. Как хранятся в памяти ЭВМ двумерные массивы?
4. Какими способами может быть описан двумерный массив?
5. Как производится ввод элементов двумерного массива?
6. Как производится вывод элементов двумерного массива?

Задания для выполнения

Часть 1.

1. Ввести двумерный массив размером m на n . Вычислить среднее арифметическое значение элементов массива, имеющих четные отрицательные значения, или вывести сообщение об отсутствии в массиве таких значений.
2. Ввести двумерный массив размером m на n . Вычислить произведение и количество элементов, находящихся в интервале от A до B , или вывести сообщение об отсутствии в массиве таких значений.
3. Ввести двумерный массив размером m на n . Вычислить среднее арифметическое значение элементов, являющихся нечётными числами, или вывести сообщение «Таких значений нет».

4. Ввести двумерный массив размером m на n . Вычислить произведение и количество элементов, кратных 3 и некратных 5, или вывести сообщение «Таких значений нет».

5. Ввести двумерный массив размером m на n . Найти и вывести те элементы, которые при делении на 7 дают остаток 2 или 5, или вывести сообщение «Таких значений нет».

6. Ввести двумерный массив размером m на n . Найти количество и сумму тех элементов, которые делятся на 5 и не делятся на 7, или вывести сообщение «Таких значений нет».

7. Ввести двумерный массив размером m на n . Вычислить произведение элементов, кратных заданному числу k (ввести с клавиатуры), или вывести сообщение «Таких значений нет».

8. Ввести двумерный массив размером m на n . Получить сумму и количество элементов, принадлежащих отрезку $[3,17]$, или вывести сообщение «Таких значений нет».

9. Ввести двумерный массив размером m на n . Найти и вывести минимальное из A_0, A_2, A_4, \dots и максимальное из A_1, A_3, A_5, \dots

10. Ввести двумерный массив размером m на n . Получить сумму тех членов массива, которые нечётны и положительны, или вывести сообщение «Таких значений нет».

11. Ввести двумерный массив размером m на n . Вычислить среднее арифметическое значение элементов, кратных сумме их индексов $(i+j)$, или вывести сообщение «Таких значений нет».

12. Ввести двумерный массив размером m на n . Вывести те значения, которые заканчиваются заданной цифрой (ввести с клавиатуры), или вывести сообщение «Таких значений нет».

13. Ввести двумерный массив размером m на n . Найти сумму тех из них, которые являются квадратами некоторого заданного числа k , или вывести сообщение «Таких значений нет».

14. Ввести двумерный массив размером m на n и некоторое число P . Вывести номера тех элементов, модуль значений которых больше этого числа, или вывести сообщение «Таких значений нет».

15. Ввести двумерный массив размером m на n , в котором есть по крайней мере два отрицательных числа. Вычислить про-

изведение и количество элементов, находящихся между первым и последним отрицательным значениями.

Часть 2.

Написать программу по обработке двумерного массива.

1. Определить количество строк, не содержащих ни одного нулевого элемента.

2. Определить количество столбцов, не содержащих ни одного нулевого элемента.

3. Определить количество столбцов, содержащих хотя бы один нулевой элемент.

4. Определить произведение элементов в тех строках, которые не содержат отрицательных элементов.

5. Определить сумму элементов в тех столбцах, которые не содержат отрицательных элементов.

6. Определить сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

7. Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

8. Найти сумму элементов в тех столбцах, которые содержат хотя бы один отрицательный элемент.

9. Найти сумму модулей элементов, расположенных ниже главной диагонали.

10. Найти сумму модулей элементов, расположенных выше главной диагонали.

11. Найти количество строк, среднее арифметическое элементов которых меньше введенной с клавиатуры величины.

12. Найти номер первой из строк, содержащих хотя бы один положительный элемент.

13. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент.

14. Найти номер первого из столбцов, не содержащих ни одного отрицательного элемента.

15. Найти номер первой из строк, не содержащих ни одного положительного элемента.