**Лабораторная работа №5**

**"Использование вложенных циклов для работы с матрицами"**

**Теория**

Матрицы – двумерные массивы. Матрицы представляют собой один из наиболее удобных математических объектов, обработка которых ведет к необходимости программирования вложенных циклов. При программной реализации обработки матриц будем использовать двумерные массивы (элементы которых имеют два индекса), считая, что элементы Хij матрицы Х, i=1, 2,...,m, j=1, 2,…,n размещаются в ячейках соответствующего массива с такими же индексами. Это позволит сделать текст программы удобным для понимания процесса обработки и уменьшить, в некоторых случаях, объём вычислений. Соответственно первый индекс массива будем называть номером строки, а второй – номером столбца.

При постановке задач на обработку матриц с целью обеспечения применимости программ для обработки матриц, например, матрицы Х, размеры которых не должны превосходить заданных значений, например, m≤12 и n≤14, будем использовать сокращенное обозначение, например, Х(m,n), m≤12, n≤14. Указанные в таком обозначении максимальные количества строк и столбцов матрицы будут использоваться в *объявлении* соответствующего *двумерного массива*, а сами эти максимальные значения желательно объявить в виде именованных констант.

При объявлении типа двумерного массива необходимо задавать количество элементов по каждому измерению, т.е. количество строк и столбцов, и тип самих элементов массива.

Задание в разделе констант максимальных значений для количества строк и столбцов улучшает стиль программирования, так как в случае необходимости изменения количества строк или столбцов программисту не придется внимательно изучать весь текст программы, а достаточно будет изменить значения соответствующих констант.

float x[mmax][nmax];

При работе с матрицами надо помнить, что первый индекс соответствует номеру строки, а второй индекс – номеру столбца.

Ввод-вывод матриц можно осуществить только с использованием вложенного цикла. Обычно принято вводить матрицу в виде матрицы, т.е. по строкам как это принято записывать на бумаге. В этом случае во внутреннем цикле должен обеспечиваться ввод (вывод) одной строки матрицы. Внешний цикл будет задавать ввод (вывод) всех строк матрицы и переход в начало новой строки после ввода (вывода) очередной строки.

Фрагмент программы, обеспечивающей ввод и вывод матрицы Х(m,n), m≤12, n≤14, имеет следующий вид:

int m,n;

printf("wwedite rasmeri m i n");

scanf("%d%d",&m,&n);

printf("\n wwedite matrizu \n");

for (int i=0;i<m;i++) //Цикл ввода всех строк матрицы

for (int j=0;j<n;j++)//Цикл ввода очередной строки матрицы

scanf("%f",&b[i][j]);

printf("\n matriza\n");

for (int i=0;i<m;i++) //Цикл вывода всех строк матрицы

{

for (int j=0;j<n;j++)//Цикл вывода очередной строки матрицы

printf("%6.1f ",b[i][j]);

printf("\n"); //Переход в начало новой строки файла вывода

}

Порядок расположения циклов при вводе-выводе имеет существенное значение. Если поменять местами заголовки циклов, то вводимые значения будут присваиваться элементам столбца, а не элементам строки. В этом случае будет выведена фактически транспонированная матрица

Также следует соблюдать аккуратность при использовании приемов программирования. Например, при вычислении сумм и произведений надо следить за тем, в каком месте программы присваиваются начальные значения соответствующим переменным.

Приведенный ниже фрагмент программы позволяет вычислить сумму и произведение всех элементов матрицы:

s=0; pr=1;

for (int i=0;i<m;i++)

for (int j=0;j<n;j++)

{

s+=a[i][j];

pr\*=a[i][j];

}

printf("\ns=%6.1f, pr=%6.1f",s,pr);

При изменении места расположения операторов присваивания s=0; pr=1**;** смысл вычислений меняется:

for(int i=0;i<m;i++)

{

s=0; pr=1;

for (int j=0;j<n;j++)

{

s+=a[i][j];

pr\*=a[i][j];

printf("\ns=%6.1f, pr=%6.1f",s,pr);

}

}

Приведенный фрагмент программы позволит вычислить сумму и произведение элементов каждой строки матрицы, а не всех элементов матрицы.

**Квадратные матрицы**

Наряду с прямоугольными матрицами часто приходится иметь дело и с квадратными матрицами, в которых количество строк равно количеству столбцов. При обработке квадратных матриц часто приходится обрабатывать элементы, расположенные в определенном месте матрицы (на главной диагонали, на побочной, под главной диагональю и т.д.). В этом случае, чтобы избежать ненужных проверок индексов элементов матрицы, целесообразно знать закон изменения индексов для элементов, расположенных в определенном месте матрицы. Для матрицы A, имеющей n строк и столбцов, можно записать следующее.

• Диагональные элементы имеют одинаковые индексы: a[i,i], i=0,n-1 .

• Элементы, стоящие над главной диагональю (образующие верхнюю треугольную матрицу), не включая диагональ: a[i,j] , i=0,n-2; j=i+1,n-1.

• Элементы, стоящие над главной диагональю, включая диагональ: a[i,j] , i=0,n-1; j=i,n-1.

• Элементы, стоящие под главной диагональю (образующие нижнюю треугольную матрицу), не включая диагональ: a[i,j] , i=1,n-1; j=0,i-1.

• Элементы, стоящие под главной диагональю, включая диагональ: a[i,j] , i=1,n-1; j=0,i.

• Элементы побочной диагонали: a[i,n-i-1], i=0,n-1.

• Элементы, стоящие над побочной диагональю, не включая диагональ: a[i,j] , i=0,n-2; j=0,n-i-2.

• Элементы, стоящие над побочной диагональю, включая диагональ: a[i,j] , i=0,n-1; j=0,n-i-1.

• Элементы, стоящие под побочной диагональю, не включая диагональ: a[i,j] , i=1,n-1; j=n-i,n-1.

• Элементы, стоящие под побочной диагональю, включая диагональ: a[i,j] , i=0,n-1; j=n-i-1,n-1.

**Практическая часть**

Пример 1. Осуществить ввод и вывод матрицы, вычисление суммы и произведения матрицы **(lab5\_1.cpp).**

Пример 2. Осуществить ввод и вывод матрицы, вычисление максимального и минимального элементов матрицы **(lab5\_2.cpp).**

Пример 3. Удалить из матрицы B(m,n), m≤10, n≤8 строки, содержащие отрицательные элементы **(lab5\_3.cpp).**

**Задание 1.** В матрице A(m,n), m≤12, n≤10, поменять местами строки с наибольшей и наименьшей суммами элементов.

Для решения поставленной задачи необходимо вычислить суммы элементов каждой строки матрицы, однако не требуется запоминать все вычисленные суммы. Поэтому после вычисления суммы элементов очередной строки матрицы следует сравнить вычисленное значение с текущими значениями максимальной и минимальной сумм и изменить, в случае необходимости, текущие значения максимума и минимума на только что вычисленное, а также запомнить номер строки, для которой эта сумма минимальна или максимальна.

После нахождения номеров строк, подлежащих обмену местами выполняется собственно обмен. Поскольку строка матрицы представляет собой одномерный массив (первый индекс имеет фиксированное значение), то эта операция выполняется с использованием обычного (не вложенного цикла), при этом обмен местами ведется с использованием дополнительной переменной, в которую будет записываться промежуточное значение при обмене.

Пример обмена значениями двух переменных a и b c использованием дополнительной переменной с:

int a=5, b=7, c;

c=a; a=b; b=c;

Таким образом, после этого фрагмента в переменной b будет записано 5, а в переменной a – 7.

**Задание 2.** Осуществить перемножение двух матриц.

В инженерной деятельности часто приходится выполнять операцию умножения матриц. Напомним, что очередной элемент результирующей матрицы C вычисляется согласно следующей формуле:

где A,B умножаемые матрицы, i, j – индексы очередного элемента матрицы C, n – количество столбцов первой матрицы (строк второй матрицы).

Перемножение матриц возможно только в том случае, если количество столбцов первой матрицы равно количеству строк второй матрицы. Умножение матриц приводит к необходимости программирования тройного цикла. Это связано с тем, что один элемент, представляющий собой сумму попарных произведений соответствующих элементов исходных матриц, вычисляется в цикле, а вычисление всех элементов матрицы требует организации еще вложенного цикла.

**Задание 3.** В квадратной матрице F(k,k), k≤9, определить, что больше: модуль минимального отрицательного элемента, стоящего над побочной диагональю, или максимальный положительный элемент, стоящий под побочной диагональю.