**Лабораторная работа на тему «Динамические массивы и матрицы»**

**Пример 1.** Выделение памяти с помощью функции calloc **(lab8\_6.cpp)**

**Пример 2.** Выделение памяти с помощью функции malloc **(lab8\_7.cpp)**

**Пример 3.** Выделение памяти с помощью функции realloc **(lab8\_8.cpp)**

**Пример 4.** Выделение памяти с помощью функции new **(lab8\_9.cpp).** Найти минимальный и максимальный элементы, и умножить каждый элемент массива на их отношение. Создать отдельные функции для ввода, вывода и преобразования массива. (Подсказка: если динамический массив создается не в функции main, то в функцию надо передавать адрес этого указателя, чтобы возвратить из функции результат, а внутри функции работа будет выполняться с указателем на указатель).

**Пример 5.** **Доступ к элементам динамической матрицы**

Для вычисления адреса элемента двумерного массива надо знать адрес первого элемента строки, в которой расположен этот элемент, и номер столбца, т.е. смещение от начала строки. Поскольку элементы многомерных массивов хранятся в памяти также в смежных ячейках, то обращение к элементам многомерных масивов можно производить как к элементам одномерного массива. Например, к элементу массива a[i][j] можно обратиться, указав два индекса, а можно, указав один индекс, при этом надо знать количество элементов в столбце: a[i\*nn+j], где . nn – количество столбцов. Такую запись может использовать в своих программах сам программист, по этой же формуле ведет вычисление адресов и компилятор.

Для доступа к элементам массивов можно использовать указатели и адресную арифметику. Как было сказано выше, адрес i-ой строки матрицы хранится в i-ом элементе массива указателей, т.е. a[i] – это и есть адрес i-ой строки матрицы. То же самое можно записать и по-другому: \*(a+i). Данная запись может трактоваться как содержимое элемента массива, имеющего смещение i от начала массива, адрес первого элеменат которого хранит переменная a.

Опираясь на адрес первого элемента строки, можно вычислять адрес и, следовательно, получать доступ к элементам этой строки. Для доступа можно использовать, во-первых, смещение относительно начала массива: \*(\*(a+i)+j). Сделаем некоторые пояснения: \*(a+i) – адрес первого элемента i-ой строки матрицы, (\*(a+i)+j) – адрес j-го элемента i-ой строки матрицы: \*(\*(a+i)+j) сам элемент, расположенный в j-ом столбце i-ой строки матрицы.

Во-вторых, чтобы сделать запись менее громоздкой, можно использовать текущий указатель на i-ую строку: p=\*(a+i); затем обратиться к нужному элементу; \*(p+j);. Разновидностью данного способа является доступ к адресу i-ой строки по ее индексу: p=a[i]; \*(p+j);.

В-третьих, можно использовать индекс для доступа к i-ой строке матрицы: \*(a[i]+j).

В-четвертых, можно использовать наиболее привычный для начинающих программистов способ, предусматривающий указание обоих индексов: a[i][j].

В примере используются различные способы доступа к элементам динамической матрицы. Сначала вычисляются элементы матрицы, а затем выводятся на монитор, при этом используются различные способы обращения к элементам матрицы. **(lab8\_10.cpp).**

**Пример 6.** Выполнить операцию вычитания двух треугольных матриц (Использование динамических массивов позволяет работать с матрицами произвольной формы, т.е. каждая строка матрицы может состоять из любого количества элементов, изменяющегося от строки к строке) **(lab8\_11.cpp).** Создать отдельные функции для ввода, вывода и вычитания матриц.

**Для всех заданий использовать динамическое выделение памяти!**

**Задание 0.** Выполнить работу с динамической матрицей: выделить память под динамическую матрицу и одномерный динамический массив, осуществить ввод элементов матрицы, а затем вычислить сумму элементов каждой строки матрицы и занести в одномерный массив.

**Задание на "Массивы":**

Задание 1. В массиве V(n), заданном начальными значениями, содержатся разные числа от 0 до 9 в произвольном порядке. Требуется поместить в массив D зашифрованную произвольную последовательность S длины L≤30 из целых чисел от 0 до 9. Шифрование выполнить по следующему правилу: Di=i-VSi. Затем по данным из D расшифровать k-тую цифру и поместить в R.

Задание 2. Из массива Х, упорядоченного по невозрастанию значений элементов, переписать в массив Y числа, исключив их повторы и обеспечив упорядоченность по возрастанию.

**Задание на "Матрицы":**

Задание 3. Составить программу, которая находит в каждой строке матрицы Q(k,l) сумму положительных элементов, расположенных между первым и последним отрицательными элементами этой же строки. Если сумму вычислить нельзя (нет положительных элементов между первым и последним отрицательными элементами), то считать ее равной нулю. Вывести исходную матрицу, располагая в одной строке элементы строки матрицы, после которых - найденную сумму, первый и последний отрицательные элементы.

Задание 4. Составить программу, которая в матрице D(m,n) находит все элементы, модуль которых располагается в интервале между средним геометрическим модулей всех элементов и средним арифметическим модулей всех элементов матрицы. Из найденных элементов сформировать одномерный массив. Вывести матрицу в виде матрицы, а под ней – элементы массива. Предполагается, что матрица нулевых элементов не содержит.