Двумерный массив

Тема. Стандартные типы данных языка программирования для представления многоэлементных однородных структур данных задачи, с бинарными отношениями между аэаементами, в программе

Цель.

- Получение навыков по определению двумерного массива для структуры данных задачи

- Получение навыков по разработке алгоритмов операций на двумерном массиве в соответствии с задачей

Оглавление

[Задание 1. Разработать программу по обработке данных, представленных в задаче матрицей и реализованной в программе двумерным (многомерным) статическим массивом. 1](#_Toc95061299)

[Задание 2. Разработать программу по обработке данных, представленных в задаче матрицей и реализованной в программе двумерным (многомерным) динамическим массивом. 2](#_Toc95061300)

[Задание 3 Разработать программу по задаче варианта с реализацией данных задачи с применением вектора библиотеки STL. 2](#_Toc95061301)

[Задание 4 Оформить отчет 3](#_Toc95061302)

[Варианты для заданий 1 и 2 3](#_Toc95061303)

[Варианты к заданию 3 5](#_Toc95061304)

[Теория и примеры по двумерным массивам 8](#_Toc95061305)

[Двумерный статический массив 9](#_Toc95061306)

[Определение двумерного статического массива 9](#_Toc95061307)

[Представление в памяти 9](#_Toc95061308)

[Доступ к элементу двумерного массива 10](#_Toc95061309)

[Указатель и двумерный массив 11](#_Toc95061310)

[Динамический двумерный массив 19](#_Toc95061311)

[Контейнер <vector> и двумерный массив 23](#_Toc95061312)

### Задание 1. Разработать программу по обработке данных, представленных в задаче матрицей и реализованной в программе двумерным (многомерным) статическим массивом.

Требования по выполнению заданий 1

1. Установить через глобальную константу максимально допустимые размеры двумерного статического массива Rows, Cols.
2. Определить массив для хранения значений целого типа (исключение для некоторых вариантов, где тип элемента определен в варианте).
3. Разработать функции по реализации алгоритмов заполнения матрицы: с клавиатуры, датчиком случайных чисел. Функция принимает на вход массив через параметр вида: Х[][Cols] и текущие размеры массива: количество строк и количество столбцов.

Например, void input(int a[][Cols], int n, int m){…….}

1. Разработать функции по реализации алгоритма вывода массива на экран построчно.
2. Определить подзадачи задачи варианта. Выполнить для них разработку реализовать, отладить.
3. Разработать и реализовать основную задачу варианта на основе функции с параметрами. Разработать тесты и выполнить тестирование.
4. Реализовать программу и отладить на тестах.

### Задание 2. Разработать программу по обработке данных, представленных в задаче матрицей и реализованной в программе двумерным (многомерным) динамическим массивом.

Требования по выполнению заданий 2

1. Ввод размеров массива для хранения данных выполнить в основной программе (функция main).
2. Двумерный массив определить как двойной указатель и выполнить его создание операцией new (или другими функциями создания переменных в динамической памяти).
3. Разработать программу задачи варианта. Используйте функции, разработанные в задании 1, возможно некоторые надо скорректировать в связи с изменением структуры представления.
4. Оформить отчет.

### Задание 3 Разработать программу по задаче варианта с реализацией данных задачи с применением вектора библиотеки STL.

Требования по выполнению задания

1. Входное множество значений хранить в структуре данных.
2. Определить структуру для хранения данных задачи. Изобразите ее для понимания представления в памяти.
3. Реализовать структуру хранения данных на основе шаблона <vector>, размеры определить при вводе с клавиатуры.
4. Выполнить декомпозицию задачи. Выявленные подзадачи реализовать через аппарат функций с параметрами.
5. Разработать функции: заполнение структуры хранения исходных данных, вывода структуры хранения.
6. Разработать и реализовать программу задачи варианта. Разработать тесты и провести тестирование задачи.
7. Определить объем памяти, требующийся программе для хранения данных.
8. Оформить отчет

### Задание 4 Оформить отчет

Структура отчета

Все задания оформить в одном документе.

В отчет по разработке задания включить:

1. Условие задачи, задание варианта, требования по выполнению
2. Постановка задачи
3. Модель решения
   1. Список подзадачи, полученных при декомпозиции с кратким описанием их алгоритма
   2. Прототипы функций подзадач
4. Алгоритмы подзадач и тесты для каждой
5. Алгоритм функции, реализующей всю задачу и тест к ней.
6. Код программы и результаты тестирования.

После отчетов по всем заданиям привести выводы о полуенных знаниях, умениях и навыках.

### Варианты для заданий 1 и 2

|  |  |
| --- | --- |
| Номер | Задача |
| 1. | Дана матрица размером n\*n. Выполнить транспонирование матриц. |
| 2. | Дана матрица размером n\*m и вектор. Умножить матрицу на вектор. |
| 3 | Даны две матрицы. Найти их сумму. |
| 4 | Даны две матрицы. Найти их разность. |
| 5 | Даны две матрицы. Найти их произведение. |
| 6 | Дана квадратная матрица. Определить, симметрична ли она относительно главной диагонали. |
| 7 | Дана квадратная матрица. Вывести ее элементы, обходя матрицу по часовой стрелке, начиная первой строки. |
| 8 | Дана квадратная матрица. Найти максимальное значение среди элементов, расположенных между главной и побочной диагоналями (значения на диагоналях не рассматриваем) |
| 9 | Дана квадратная матрица. Найти минимальное значение среди элементов, расположенных под главной диагональю. |
| 10 | Дана квадратная матрица. Найти минимальное значение среди элементов, расположенных над побочной диагональю. |
| 11 | Дана квадратная матрица. Найти минимальное значение среди элементов, расположенных под побочной диагональю. |
| 12 | Дана прямоугольная матрица. Сформировать массив индексов седловых точек матрицы. Седловой точкой назовем элемент, который является наибольшим в своей строке и наименьшим в столбце. |
| 13 | Дана прямоугольная матрица. Переместить минимальный элемент матрицы в ее левый верхний угол, путем перестновки строк и столбцов. |
| 14 | Дано множество точек на плоскости, точки заданы своими координатами. Найти наибольшее расстояние между этими точками. |
| 15 | Дана квадратная матрица. Найти в диагональ, среди параллельных главной, сумма значений которой минимальна. |
| 16 | Дана квадратная матрица. Сформировать одномерный массив произведений элементов главной и побочной диагоналей, имеющих одинаковый индекс строки. |
| 17 | Даны три вещественные прямоугольные матрицы А, В, С размером n\*m каждая. Вычислить величину  Где запись вида |D| обозначает алгоритм вычисления значения  |D|=max(первой строки)+max(второй строки)+…+max(n-ой строки) |
| 18 | Дана матрица размером n\*m и номер строки. Сформировать массив и включить в его i-ый элемент значение, равное сумме максимального и минимального элемента i-ой строки исходной матрицы. |
| 19 | Шахматная доска 8\*8. Вводятся координаты ферзя. Отметить клетки (вставить какое-то значение) поля доски, которые находятся под боем ферзя. Обращение к полям доски выполнять по правилам игры в шахматы |
| 20 | Дана целочисленная квадратная матрица. Определить: 1) сумму элементов в тех столбцах, которые не содержат отрицательных элементов  2) минимум среди сумм модулей элементов диагоналей, параллельных побочной диагонали матрицы. |
| 21 | Дана целочисленная квадратная матрица порядка n. Найти номера строк, элементы каждой из которых образуют монотонную последовательность (монотонно убывающую или монотонно возрастающую). |
| 22 | Клеточное поле размером m \* n является результатом игры в крестики-нолики. Проверить, не закончена ли игра выигрышем "крестиков". Выигрыш наступает при образовании цепочки по горизонтали, вертикали или диагонали из 5 крестиков подряд. |
| 23 | Дана квадратная матрица. Найти определитель данной матрицы методом Гаусса. |
| 24 | Найдите матрицу, обратную данной матрицы методом Гаусса. |
| 25 | Дана матрица размером n\*m. Операция сглаживания матрицы дает новую матрицу того же размера, каждый элемент которой получается, как среднее арифметическое соседей соответствующего элемента исходной матрицы. Построить результат сглаживания заданной матрицы. |
| 26 | Дана матрица a размером n\*n, заполненная неотрицательными целыми числами. Расстояние между двумя элементами aij и apq определено как |i−p|+|j−q|. Требуется заменить каждый нулевой элемент матрицы ближайшим ненулевым. Если есть две или больше ближайших ненулевых ячейки, нуль должен быть оставлен. |
| 27 | Решить систему линейных алгебраических уравнений методом Холецкого. |
| 28 | Дана квадратная матрица, состоящая из натуральных чисел. Зеркально отразить (транспонировать) ее элементы относительно побочной диагонали. |
| 29 | Дана квадратная матрица. Проверить, что в данной матрице произведение элементов, стоящих над побочной диагональю, равно произведению элементов, стоящих над главной диагональю |
| 30 | Дана матрица размером n\*m элементы которой заполнены цифрами от 0 до 9. Требуется найти такой путь из клетки (1,1) в клетку (n, m), чтобы сумма цифр в клетках, через которые он пролегает, была минимальной; из любой клетки ходить можно только вниз или вправо. |

### Варианты к заданию 3

|  |  |
| --- | --- |
| Номер варианта | Задача |
| 1 | Два выпуклых многоугольника на плоскости заданы координатами вершин в порядке обхода границы. Определить площадь многоугольника и определить, вложены ли они. |
| 2 | Из заданного на плоскости множества точек выбрать три различные точки так, чтобы разность между площадью круга, ограниченного окружностью, проходящей через эти три точки, и площадью треугольника с вершинами в этих точках была минимальной. |
| 3 | Даны два множества точек на плоскости. Выбрать три различных точки первого множества так, чтобы круг, ограниченный окружностью, проходящий через эти три точки содержал все точки второго множества и имел минимальную площадь. |
| 4 | Даны два множества точек на плоскости. Выбрать четыре различных точки первого множества так, чтобы квадрат с вершинами в этих точках включал все точки второго множества. |
| 5 | Даны два множества точек на плоскости. Выбрать три различных точки первого множества так, чтобы треугольник с вершинами в этих точках, содержал все точки второго множества и имел минимальную площадь. |
| 6 | Даны два множества точек на плоскости. Найти радиус и центр окружности, проходящей через n (n>=3) точек первого множества и содержащей строго внутри себя разное число точек первого и второго множества. |
| 7 | Даны два множества точек на плоскости. Из первого множества выбрать три различные точки так, чтобы треугольник с вершинами в этих точках, содержал (строго внутри себя) равное количество точек первого и второго множества. |
| 8 | На плоскости задано множество точек М и круг. Выбрать из М две различные точки так, чтобы наименьшим образом различались количества точек в круге, лежащие по разные стороны от прямой, проходящей через эти две точки. |
| 9 | Дано 3n точек на плоскости, причем никакие три из них не лежат на одной прямой. Построить множество n треугольников с вершинами в этих точках так, чтобы никакие два треугольника не пересекались и не содержали друг друга. |
| 10 | Выбрать три различных точки из множества точек на плоскости так, чтобы была минимальной разность между количеством точек, лежащих внутри и вне треугольника с вершинами в выбранных точках. |
| 11 | Определить радиус и центр окружности, проходящей по крайней мере через три различные точки заданного множества точек на плоскости и содержащей внутри наибольшее количество точек этого множества. |
| 12 | На плоскости заданы множество точек А и точка d вне его. Подсчитать количество различных неупорядоченных троек точек a, b, c из А таких, что четырехугольник abcd является параллелограммом. |
| 13 | На плоскости заданы множество точек А и множество окружностей В. Найти две такие различные точки из А, что проходящая через них прямая пересекается с максимальным количеством окружностей из В. |
| 14 | Задано множество точек на плоскости. Найти все четверки точек, являющихся вершинами квадратов. Найти квадрат, внутри которого лежит наибольшее количество точек множества. |
| 15 | Определить радиус и центр окружности минимального радиуса, проходящего через три различные точки заданного множества точек на плоскости. |
| 16 | Найти три треугольника с вершинами в заданном множестве точек на плоскости так, чтобы второй треугольник лежал строго внутри первого, а третий внутри второго. |
| 17 | Дано множество точек на плоскости. Построить все возможные треугольники с вершинами в этом множестве точек и найти среди них такой, стороны которого пересекаются с максимальным количеством треугольников. |
| 18 | На плоскости заданы множества точек и окружность радиусом R с центром в начале координат. Построить множество всех треугольников с вершинами в заданных точках, все три стороны которого пересекаются с окружностью, и найти среди них треугольник с минимальной стоимостью. |
| 19 | Подсчитать количество равносторонних треугольников с различными длинами оснований и вершинами в заданном множестве точек на плоскости и определить, пересекаются ли они. |
| 20 | Множество попарно различных плоскостей в трехмерном пространстве задано перечислением трех точек, через которые проходит каждая из плоскостей. Выбрать максимальное подмножество попарно непараллельных плоскостей. |
| 21 | На плоскости задано множество точек с целочисленными координатами. Необходимо найти количество отрезков, обладающих следующими свойствами:  1) оба конца отрезка принадлежат заданному множеству;  2) ни один конец отрезка не лежит на осях координат;  3) отрезок пересекается ровно с одной осью координат.  Напишите эффективную по времени и по используемой памяти программу для решения этой задачи. |
| 22 | Найти решение системы линейных алгебраических уравнений методом Гаусса. |
| 23 | Текст вводится как последовательность символов, заканчивающаяся точкой. Определить количество различных пар букв в этом тесте. |
| 24 | Дано множество из пятнадцати точек на плоскости. Выяснить верно ли, что для каждой из этих пятнадцати точек найдется другая, что все остальные тринадцать точек лежат по одну сторону от прямой, проходящей через эти точки. |
| 25 | Дано множество из n (n – нечетное) точек на плоскости. Найти число медиан этого множества точек на плоскости в предположении, что никакие три точки не лежат на одной прямой.  *Медианой множества, состоящего из четного числа точек на плоскости, никакие три из которых не лежат на одной прямой, называется прямая, которая делит множество на два подмножества одинаковой мощности.* |
| 26 | Дана система линейных алгебраических уравнений АХ=В. Найти решение системы из n линейных уравнений, рассматривая ее как треугольную. Пример треугольной системы с четырьмя переменными.  A11x1 + A12x2 + A13x3 + A14x4 = b1  A22x2 + A23x3 + A24x4 = b2  A33x3 + A34x4 = b3  A44x4 = b4  Сформировать вектор решений. При подготовке тестов привести систему к треугольному виду вручную. |
| 27 | Дано множество точек на плоскости. Построить симметричное ему множество, относительно прямой Ах+Ву+С=0. |
| 28 | Дано множество из n точек на плоскости. Найти горизонтальную медиану этого множества точек на плоскости в предположении, что никакие две точки не лежат на одной горизонтальной прямой.  *Медианой множества точек на плоскости назовем прямую, которая делит множество на два подмножества одинаковой мощности.* |
| 29 | Дано множество из n точек на плоскости. Найти вертикальную медиану этого множества точек на плоскости в предположении, что никакие две точки не лежат на одной вертикальной прямой.  *Медианой множества точек на плоскости назовем прямую, которая делит множество на два подмножества одинаковой мощности.* |
| 30 | Дано конечное множество точек на плоскости. Нужно найти окружность минимального радиуса такую, чтобы данные точки были внутри неё. |

Определение. Медианой множества, состоящего из четного числа точек плоскости, никакие три из которых не лежат на одной прямой, называется прямая, соединяющая две точки множества, с обеих сторон от которой лежит равное число точек.  
Даны действительные числа x1, y1, x2, y2, …, xn, yn (n – нечетное число). Найти число медиан множества точек с координатами (x1, y1),( x2, y2),…( xn, yn) в предположении, что никакие три точки этого множества не лежат на одной прямой.

### Теория и примеры по двумерным массивам

Структура данных – это совокупность элементов, связанных друг с другом некоторыми отношениями. Для представления таких данных в программе, используются структуры хранения в оперативной памяти, они хранят значения элементов и отношения между элементами. Элементом структуры хранения является ячейка памяти, предназначенная для хранения значения, а способ соединения ячеек между собой определяет вид структуры хранения: линейная или нелинейная.

В программе структуру хранения представляют стандартные структурные типы данных: массив, строка, структура, объединение, перечисление и типы более сложные, определяемые пользователем.

Массив используется для представления в программе сложных структур данных. Массив - это линейная, однородная структура данных с прямым доступом к элементу, состоящая из конечного числа элементов.

Одномерный массив предназначен для представления структур данных, элементы которых подчиняются линейным отношениям между элементами, элементы размещаются в массиве в соответствии с порядковым номером. Например, требуется обрабатывать в программе среднесуточную температуру за январь месяц, и находить температуру в заданный день января.

Двумерный массив предназначен для представления структур данных элементы которых подчиняются бинарным (2-арным)отношениям. Каждый элемент структуры данных располагается в массиве в ячейке в соответствии с двумя параметрами (их еще называют координатами)- номер строки и номер столбца. Например, для хранения матрицы размером 3×4 очень удобно использовать двумерный массив.

### Двумерный статический массив

### Определение двумерного статического массива

*Без инициализации*

[КП] ***имя типа*** **имя массива**[*кол-во строк*][*кол-во столбцов*]

*Имя типа* – любой стандартный тип языка Си или пользовательский тип, включая пользовательский и указатель.

*кол-во строк –* количество строк в массиве.

*кол-во столбцов –* количество столбцов в строке.

Пример 1. Определение глобального двумерного массива размером 2×3 и локального 3×3 без инициализации.

#include “stdio.h”

int x[2][3]; //массив из 2 строк и 3 столбцов

int main()

{

float y[3][3]; //массив из 3 строк и 3 столбцов

}

Пример 2. Определение глобального двумерного массива размером 2×3 и локального 3×3 с инициализацией тремя способами.

#include “stdio.h”

#define ROWS 2

#define COLS 2

int x[2][3]={1,2,3,4,5,6}; //перечисление всех значений

int main()

{

float y[3][3]={{1,2,3},{5,6,7},{4,11,9}}; //*перечисление значений строк*

float z[][3]={{1,2,3},{5,6,7},{4,11,9}}; //*в определении массива параметр количество строк можно опустить, но количество столбцов опускать нельзя.*

int a[ROWS][COLS];

}

### Представление в памяти

Определение float y[3][3]; можно разделить на составные части: float y[3] – можно прочитать так: у – указатель на массив из трех элементов, а вторая скобка [3] вновь указывает на массив из трех элементов – т.е. снова на массив (а массив и указатель это одно и то же).

В язык Си имя двумерного массива это указатель на массив указателей на строки. Ниже приведена структура **физического представления** в памяти двумерного массива y.



Логически массив можно ассоциировать с матрицей. Например, массив ***y*** можно представить матрицей вида:

|  |  |
| --- | --- |
|  | Где строка матрицы – это строка двумерного массива.  При разработке программ обработки матриц, для их хранения и используют двумерный массив. |

### Доступ к элементу двумерного массива

Осуществляется через переменную, имя которой определяется по формату:

***Имя массива***[*индекс строки*][*индекс столбца*]

Индексы строк и столбцов начинаются ***с нуля***.

Пример 3 Увеличение значений элементов первой (нулевой) строки массива **y** в двое

int main()

{

float y[3][3]={{1,2,3},{5,6,7},{4,11,9}};

y[0][0]= y[0][0]\*2;

y[0][1]= y[0][1]\*2;

y[0][2]= y[0][2]\*2;

}

Пример 4 заполнение двумерного массива с клавиатуры построчно.

#include “stdio.h”

#define n 2

#define m 3

int main()

{

int x[n][m];

int i, j;

printf(“ Введите % чисел”, n\*m);

for(i=0;i<n;i++)

{

for(j=0;j<m;j++) //заполнение элементов i-ой строки

{

scanf(“%d”,&x[i][j]);

}

}

}

Варианты подготовки входных данных

1. Все шесть чисел через пробел

12 4 15 6 7 2

1. В две строки

12 4 15

6 7 2

Пример 5 вывод двумерного массива на монитор построчно.

#include “stdio.h”

#define n 2

#define m 3

int main()

{

int x[n][m];

int i, j;

printf(“ Введите % чисел”, n\*m);

for(i=0;i<n;i==)

{

for(j=0;j<m;j++) //вывод i-ой строки

{

printf(“%d ”,x[i][j]);

}

printf(“\n”); //переход на другую строку

}

}

### Указатель и двумерный массив

Имя двумерного массива – это указатель на массив указателей. Двумерный массив можно рассматривать как одномерный массив, элементы которого - одномерные массивы.

Пример 6. Присваивание двумерного массива х указателю у и обработка двумерного массива у, перебором элементов как в одномерного массива.

int \_tmain(int argc, \_TCHAR\* argv[])

{

int x[2][3]={{1,2,3},{4,5,6}};//x – массив из двух указателей на массивы

//из трех элементов

int \*y; //просто указатель – т.е. одномнрный массив

y=\*x;

int i,j;

for(i=0;i<2;i++)

{

for(j=0;j<3;j++)

{

printf("%d",y[i\*3+j]);

}

printf("\n");

}

return 0;

}

Пример 7. Присваивание двумерного массива массиву указателей. Обработка двумерного массива чрез массив указателей – **p**

int main(int argc, char\* argv[])

{

int \*p[2], //p- массив из двух указателей на int

a[2][3]={{1,5,3},{4,5,6}} //а – указатель на массив из двух

// указателей на массив из трех элементов

p=a;

for(int i=0;i<3;i++)

{

p[i]=a[i];

for(int j=0;j<2;j++)

cout<<p[i][j]; //доступ к элементам через р

}

return 0;

}

Передача двумерного массива в функцию

Пример 7. Заполнение двумерного массива с клавиатуры и вывод на монитор

#include "stdafx.h"

#include "iostream"

#define cols 10

#define rows 10

using namespace std;

void inp\_arr(int x[][cols],int n, int m)

{

int i,j;

for(i=0;i<n;i++)

{

for(j=0;j<m;j++)

{

scanf("%d",&x[i][j]);

}

}

}

void out\_arr(int x[][cols],int n, int m)

{

int i,j;

for(i=0;i<n;i++)

{

for(j=0;j<m;j++)

{

printf("%d",x[i][j]);

}

}

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int x[rows][cols], n,m;

printf("Введите количество строк в массиве");

scanf("%d",&n);

printf("Введите количество элементов в строке массива");

scanf("%d",&m);

if (n<=0 || m<=0)

{

printf("Количество строк и столбцов >0");

return 1;

}

inp\_arr(x,n,m);

out\_arr(x,n,m);

getchar();

getchar();

return 0;

}

Пример 8. Дана квадратная матрица. Определить симметрична ли матрица относительно главной диагонали. (На главной диагонали матрицы находятся элементы, индексы строки и столбца которых одинаковые.)

***j***

 элементы y***ij***

***i*** Главная диагональ: индексы равны i=j

Решение: строки ***i*** проходим все, а столбцы ***j*** в сроке только до главной диагонали.

int sim\_arr(int x[][3],int n) //возвращает 1 или 0

{

int i, j;

for(i=0;i<n;i++)

{

for(j=0;j<i;j++) //до главной диагонали

{

if (x[i][j]!=x[j][i]) return 0;

}

}

return 1;

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int a[3][3]={1,2,1,

1,1,1,

1,1,1};

if (sim\_arr(a,3))

cout<<"Y";

else

cout<<"N";

getchar();

getchar();

return 0;

}

Пример 9. Предача двумерного массива через указатель.

Увеличить все значния двумерного массива вдвое.

// Matrica.cpp: определяет точку входа для консольного приложения.

//

#include "stdafx.h"

#include "iostream"

#define cols 5

#include "locale.h"

using namespace std;

void out\_matr(int matr[][3],int n, int m)//вывод двумерного массива

{

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

{

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

cout<<matr[i][j]<<' ';

cout<<'\n';

}

}

//умножение значений элементов, передача двумерного массива

void umn\_matr(int matr[][3],int n, int m)//умножение значений элементов

{

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

{

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

matr[i][j]\*=2;

}

}

//умножение значений элементов, передача двумерного массива как одномерного

void umn\_matr1(int \*matr,int n, int m)

{

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

{

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

matr[i\*m+j]\*=2; //формирование индекса

}

}

//вывод переданной сроки двумерного массива

void out\_arr(int \*x, int m)

{

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

cout<<x[j]<<' ';

cout<<'\n';

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int a[2][3]={1,2,3,4,5,6};

setlocale(LC\_ALL,"rus");

cout<<"Вывод строки"<<'\n';

out\_arr(a[0],3);//передача строки матрицы

cout<<"Матрица до модификации"<<'\n';

out\_matr(a, 2,3);// передача матрицы

umn\_matr1(\*a, 2,3);// передача матрицы как одномерного массива

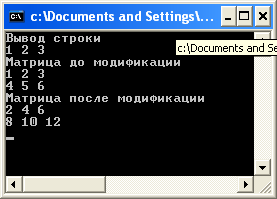
cout<<"Матрица после модификации"<<'\n';

out\_matr(a, 2,3);// передача матрицы

return 0;

}

Результат



Передача строк двумерного массива в функцию

Пример 10.Дан двумерный массив целых чисел. Сформировать одномерный массив, каждый элемент которого наибольший общий делитель соответствующей строки двумерного массива.

Например, для двумерного массива

3 6 9 12

2 4 8 16

5 10 15 20

Решением будет массив: 3 2 5

Декомпозиция

1. Найти НОД двух целых чисел



int NOD(int a,int b)

//предусловие: a и b одновременно не могут быть равны нулю

1. Найти НОД элементов одномерного массива из n целых чисел



int NOD\_arr(int \*x,int n)

//предусловие: n должно быть >=1

1. Сформировать массив нодов строк двумерного массива



void create\_arr(int a[][4],int n,int m, int \*x)

//предусловие: n,m должно быть >=1

//количество эементовв х равно n

1. Вывод одномерного массива

void out\_arr (int\*x,int n)

Реализация программы

#include "stdafx.h"

#include "iostream"

#define cols 10

#define rows 10

using namespace std;

int NOD(int a,int b)//a и b одновременно не могут быть равны нулю

{

if(a==0||b==0)return a+b;

else

{

while(a!=b)

{

if(a>b)a=a-b;

else

b=b-a;

}

return a;

}

}

int NOD\_arr(int \*x,int n) //n должно быть >=1

{

int i,a=x[0],b;

for(i=1;i<n; i++)

{

b=x[i];

a=NOD(a,b);

}

return a;

}

void create\_arr(int a[][4],int n,int m, int \*x)

{

int i,j;

for(i=0;i<n;i++)

{

x[i]=NOD\_arr(a[i],m);

}

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int a[3][4]={3,6,9, 12, //исходный двумерный массив

2,4,8, 16,

5,10,15,20};

int x[3]; //результат

create\_arr(a,3,4,x);

getchar();

getchar();

return 0;

}

### Динамический двумерный массив

int \*\*x;

Такая запись называется косвенным указателем, т.к. х является указателем на ячейку, которая так же является указателем.

Двойной указатель используется для определения динамического двумерного массива.

Пример 11. Определение двойного указателя:

int main()

{

double a=5;

double \*d;

double \*\*pd;

d=&a;

pd=&d;

cout<<\*d; //5

cout<<\*\*pd; //5

}



**Применение функции malloc и free**

Пример 11. Создание динамического массива и перестановка строк массива с использованием указателей на строку

#include "stdafx.h"

#include "iostream"

#define cols 5

#include "locale.h"

using namespace std;

//вывод двумерного статического массива

void out\_matr(int matr[][3],int n, int m)

{

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

{

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

cout<<matr[i][j]<<' ';

cout<<'\n';

}

}

//вывод двумерного динамического массива

void out\_matr(int \*\*matr,int n, int m)

{

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

{

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

cout<<matr[i][j]<<' ';

cout<<'\n';

}

}

//перестановка строк в двумерном статическом массиве

void cheng\_matr(int matr[][3],int m, int k, int l)

{

int tmp,j;

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

{

tmp=matr[k][j];

matr[k][j]=matr[l][j];

matr[l][j]=tmp;

}

}

//создание динамического двумерного массива функцией malloc

int\*\* create\_din\_dvum(int n,int m)

{

int \*\*b=(int \*\*) malloc(sizeof(int)\*n);

int i,j;

for(i=0;i<n;i++)

{

b[i]=(int \*) malloc(sizeof(int)\*m);

for(j=0;j<m;j++)

{ b[i][j]=rand()/100;

}

}

return b;

}

//престановка строк c номерами k и l в двумерном **динамическом** массиве

void cheng\_matr(int \*\*matr,int m, int k, int l)

{

int tmp,j;

int\* p;

p=matr[k];

matr[k]=matr[l];

matr[l]=p;

}

int \_tmain(int argc, \_TCHAR\* argv[])

{

int a[2][3]={1,2,3,4,5,6};

setlocale(LC\_ALL,"rus");

cout<<"Вывод строки"<<'\n';

out\_arr(a[0],3);//передача строки матрицы

cout<<"Матрица до модификации"<<'\n';

out\_matr(a, 2,3);// передача матрицы

umn\_matr1(\*a, 2,3);// передача матрицы как одномерного массива

cout<<"Матрица после модификации"<<'\n';

out\_matr(a, 2,3);// передача матрицы

cheng\_matr(a,3, 0, 1);

cout<<"Матрица после перестановки строк"<<'\n';

out\_matr(a, 2,3);// передача матрицы

//Создание динамического двумерного массива

int \*\*B=create\_din\_dvum(4,3);

cheng\_matr(B,3, 0, 3);

cout<<"Матрица после перестановки строк"<<'\n';

out\_matr(B, 4,3);

return 0;

}

**Применение опрации new и delete**

Пример 12. Создание двухмерного массива из n\*m элементов. Вывод элементов второй строки .Массив создается как одномерный. Элементы адресуются с помощью указателя на массив.

int main(int argc, char\* argv[])

{

int n,m,\*arr;

cin>>n>>m;

arr=new int[n\*m];

for (int i=0;i<n\*m;i++) cin>>\*(arr+i);

// вывод элементов второй строки

int j=2;

for ( i=0;i<n;i++) cout<<\*(arr+i+(j-1)\*n);

delete [] arr;

}

Пример 10. Создание двухмерного массива через массив указателей на строки массива. Число строк и длина строки известны.

int main(int argc, char\* argv[])

{

int n,m;

cin>>n>>m;

int \*\*arr=new int \*[n]; // создание массива из n указателей

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

arr[i]=new int[m]; // создание n строк из m элементов

// каждая

for ( i=0;i<n;i++)

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

cin>>arr[i][j];

for ( i=0;i<n;i++)

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

cout<<arr[i][j];

//удаление двумерного динамического массива

for ( i=0;i<n;i++) //Удаление строк

delete [] arr[i];

delete [] arr; //удалении массива указателей

}

Пример 11. Создание двухмерного массива через массив указателей на строки массива. Число строк известно. Длина строки переменная.

int main(int argc, char\* argv[])

{

int n,m,j;

cin>>n;

int \*\*arr=new int \*[n];

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

{

cin>>m;

arr[i]=new int[m+1]; //резервируем доп.ячейку под размер строки

arr[i][0]=m; // запоминается длина строки в ячейке с номером 0

}

//вывод и удаление массива

for ( i=0;i<n;i++)

{

for ( int j=1;j<=arr[i][0];j++)

cout<<arr[i][j];

for ( i=0;i<n;i++)

delete [] arr[i];

}

delete arr;

return 0;

}

### Контейнер <vector> и двумерный массив

Если считать, что одна строка — это **vector**<int>, то **двумерный** **массив** — это **вектор** элементов типа **vector**<int>, то есть его нужно объявлять как **vector**<**vector**<int> >. При этом по стандарту языка C++ до 2011 года, в конце определения между двумя символами “<” должен стоять пробел, начиная с C++11 этот пробел необязателен.

Когда вектор объявляется внутри другого вектора, этот вектор называется 2-мерным вектором, который работает как 2-мерный массив. Двумерный вектор содержит несколько строк, каждая из которых является другим вектором. В этом руководстве показано использование двумерного вектора в C ++.

Пример создания двумерного вектора размером n\*m при его заполнении (при добавлении нового элемента в конец массива)

int main() {

int n=4, m=5;

vector<vector<int>> vec;

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

{

vector <int> temp;

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

{

temp.push\_back(rand()%10);

}

}

cout << endl;

for (int vecline = 0; vecline < 3; vecline++)

{

for (int vecelem = 0; vecelem < 5; vecelem++)

{

cout << vec[vecline][vecelem] << " ";

}

cout << endl;

}

return 0;

}

Пример создания свободного и отображения двумерного массива на основе вектора. Реализация на функции с параметром-вектором и возврат созданной структры.

//свободный двумерный массив на основе вектора

vector<vector<int>> create2Dvector(int row) {

int col;

//Resize the outer vector

vector<vector<int>> X;

X.resize(row);

int val;

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

{

cout << "Введите размер строки " << i << endl;

cin >> col;

X[i].resize(col);

cout << "Введите " << col << "чисел для заполнения строки " << i << endl;

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

{

//Take input from the user

cin >> val;

//Insert into the vector

X[i][j] = val;

}

}

return X;

}

//Print the values of the vector

void print2Dvector(vector<vector<int>> X) {

cout << "The values of the vector are : \n";

for (unsigned int i = 0; i < X.size(); i++)

{

for (unsigned int j = 0; j < X[i].size(); j++)

cout << X[i][j] << " ";

cout << '\n';

}

}Пример создания двумерного массива со строками равной длины на основе вектора и его инициализация при создании

vector<vector<int>> intVector

{ {20,10}, {10, 30}, {50, 40 }, {80, 10 } };

//Print the values of the vector using for loop

cout << "The values of the vector are : \n";

for (vector<int> row : intVector)

{

for (int val : row)

cout << val << " ";

cout <<'\n';

}

Пример создания двумерного массива со строками различной длины (свободный массив) длины на основе вектора и его инициализация при создании

vector<vector<int>> intVector

{ {20}, {10, 30}, {50, 40, 60 }, {80, 10, 70, 9, 0 } };

//Print the values of the vector using for loop

cout << "The values of the vector are : \n";

for (vector<int> row : intVector)

{

for (int val : row)

cout << val << " ";

cout <<'\n';

}