**МИНЕСТРЕСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Балтийский государственный технический университет «ВОЕНМЕХ»

им. Д.Ф. Устинова»

Кафедра О7

«Информационные системы и программная инженерия»

Практическое задание № 6

по дисциплине «Информационные технологии и программирование»

на тему «Двумерные массивы»

Выполнила:

Студент(ка) Карпинская Полина Анатольевна

Группа Е721б

Преподаватель: Удовиченко Андрей Сергеевич,

канд. физ.-мат. наук, доцент

Санкт-Петербург

2023

**Цель работы** – освоить работу с матрицами, научиться правильно организовывать циклы для обработки многомерных массивов.

Скрипт practical\_task\_6\_1.c:

#include <stdio.h>

#define n 9

#define m 12

int inputMatrix(int Y[n][m]) {

int flag = 1;

printf("Enter the array elements:\n");

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

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

if (scanf("%d", &Y[i][j]) != 1 && getchar() != '\n') flag = 0;

}

}

return flag;

}

void printMatrix(int Y[n][m]) {

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

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

if (n > j-1) printf("%d ", Y[i][j]);

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

}

printf("\n");

}

printf("\n");

}

int main() {

int Y[n][m];

if (inputMatrix(Y) == 1) {

int Sn = 0, So = 0;

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

if (i % 2 != 0) {

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

if (Y[i][j] > 0) Sn += Y[i][j];

}

} else {

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

if (Y[i][j] < 0) So += Y[i][j];

}

}

}

float F = (float)(Sn + So) / (Sn - So);

printf("F = %.2f\n", F);

} else printf("Error! Incorrect values entered!");

return 0;

}

Скрипт practical\_task\_6\_2.c:

#include <stdio.h>

#define n 10

#define m 10

#define size (n \* m)

int inputMatrix(int A[n][m]) {

int flag = 1;

printf("Enter the array elements:\n");

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

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

if (scanf("%d", &A[i][j]) != 1 && getchar() != '\n') flag = 0;

}

}

return flag;

}

void printMatrix(int A[n][m]) {

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

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

if (n > j-1) printf("%d ", A[i][j]);

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

}

printf("\n");

}

printf("\n");

}

void sortDiagonal(int A[n][m]) {

int diagonal[size];

int index = 0;

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

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

if (i == j) {

diagonal[index] = A[i][j];

index++;

}

}

}

for (int i = 0; i < index - 1; i++) {

for (int j = 0; j < index - i - 1; j++) {

if (diagonal[j] > diagonal[j + 1]) {

int temp = diagonal[j];

diagonal[j] = diagonal[j + 1];

diagonal[j + 1] = temp;

}

}

}

index = 0;

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

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

if (i == j) {

A[i][j] = diagonal[index];

index++;

}

}

}

}

int main() {

int A[n][m];

inputMatrix(A);

sortDiagonal(A);

printf("The matrix after sorting:\n");

printMatrix(A);

return 0;

}

Ответы на контрольные вопросы:

1. Матрица - это упорядоченный двумерный массив элементов, состоящий из строк и столбцов.

2. Матрица располагается в памяти компьютера последовательно, построчно или постолбцово, в зависимости от выбранного формата хранения.

3. Матрица может быть проинициализирована путем присваивания значений элементам во время объявления или путем последующей их инициализации через циклы или ввод с клавиатуры.

4. При объявлении матрицы с одновременной инициализацией первые скобки можно оставить пустыми, так как компилятор самостоятельно определит их размеры на основе предоставленной инициализации, в то время как вторые скобки обязательны для указания размеров столбцов.

5. При обращении к элементу матрицы сначала указывается индекс строки, а затем индекс столбца.

6. Для вывода матрицы таблицей с колонками одинаковой ширины можно использовать форматированный вывод с заданием ширины столбцов.

7. Имя матрицы представляет собой уникальное идентификационное имя, по которому можно обращаться к матрице для выполнения операций.

8. Индекс строки матрицы представляет собой целое число, указывающее номер строки, начиная с 0.

9. Индекс столбца матрицы представляет собой целое число, указывающее номер столбца, начиная с 0.

10. Имя строки матрицы представляет собой уникальное идентификационное имя, по которому можно обращаться к определенной строке матрицы.

11. Для получения адреса элемента матрицы можно использовать указание индексов строки и столбца, в котором он находится.

12. Адрес начала матрицы представляет собой адрес первого элемента матрицы.

13. Адрес строки матрицы представляет собой адрес первого элемента заданной строки матрицы.

14. Просмотр элементов матрицы по столбцам отличается от просмотра по строкам тем, что при просмотре по столбцам мы фиксируем столбец и перемещаемся по строкам, а при просмотре по строкам мы фиксируем строку и перемещаемся по столбцам.

15. Индексы строки и столбца у элементов, лежащих на главной диагонали квадратной матрицы, всегда одинаковые. Например, элемент с индексами (i, i) лежит на главной диагонали и его значение зависит от значения индекса i.

16. Индексы строки и столбца у элементов, лежащих на побочной диагонали квадратной матрицы, связаны следующим образом: для элемента с индексами (i, j), где i - номер строки, j - номер столбца, сумма i и j будет равна размерности матрицы минус 1.

17. Да, можно просмотреть все элементы матрицы в одном цикле. Для этого можно использовать два вложенных цикла, один для перемещения по строкам, и второй для перемещения по столбцам.

18. Чтобы обратиться к некоторому элементу матрицы через указатель на первый элемент матрицы, нужно знать размерность матрицы и используя арифметику указателей вычислить нужный элемент. Например, для элемента с индексами (i, j) в матрице размером n x m можно использовать указатель на первый элемент и подсчитать смещение по формуле: offset = i \* m + j.

19. Чтобы обратиться к элементу матрицы через указатель на начало матрицы, нужно знать размерность матрицы и используя арифметику указателей вычислить нужный элемент. Например, для элемента с индексами (i, j) в матрице размером n x m можно использовать указатель на начало матрицы и подсчитать смещение по формуле: offset = i \* m + j.

20. Трехмерный массив - это массив, состоящий из элементов, у которых есть три индекса: первый - номер слоя, второй - номер строки, третий - номер столбца. Для работы с трехмерным массивом используются три вложенных цикла, один для перемещения по слоям, второй для перемещения по строкам, и третий для перемещения по столбцам. Все операции над трехмерным массивом аналогичны операциям над двумерным массивом, только нужно учитывать наличие третьего индекса при обращении к элементам.