МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное образовательное учреждение высшего образования

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ

УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ОТЧЕТ О ВЫПОЛНЕНИИ

ЛАБОРАТОРНОЙ РАБОТЫ №4

«Двумерные массивы»

Студента 2 курса, группы ИСП-207

Асылбек уулу Бакыта

Направление 09.02.07 – «Информационные системы и программирование»

Руководитель:

преподаватель

М.В. Валеев

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Работа защищена

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_2022г.

Москва, 2022

Оглавление

[Задача №1 4](#_Toc104028374)

[1. Постановка задачи 4](#_Toc104028375)

[2. Структурные требования 4](#_Toc104028376)

[3. Описание задачи 4](#_Toc104028377)

[3.1. Используемые переменные 4](#_Toc104028378)

[3.2. Псевдокод 4](#_Toc104028379)

[3.3. Блок-схема 4](#_Toc104028380)

[3.4. Листинг программной реализации 7](#_Toc104028381)

[3.5. Результаты 8](#_Toc104028382)

[Задача №2 9](#_Toc104028383)

[1. Постановка задачи 9](#_Toc104028384)

[2. Структурные требования 9](#_Toc104028385)

[3. Описание задачи 9](#_Toc104028386)

[3.1. Используемые переменные 9](#_Toc104028387)

[3.2. Псевдокод 9](#_Toc104028388)

[3.3. Блок-схема 10](#_Toc104028389)

[3.4. Листинг программной реализации 12](#_Toc104028390)

[3.5. Результаты 13](#_Toc104028391)

[3.6. Тесты 14](#_Toc104028392)

[Задача №3 15](#_Toc104028393)

[1. Постановка задачи 15](#_Toc104028394)

[2. Структурные требования 15](#_Toc104028395)

[3. Описание задачи 15](#_Toc104028396)

[3.1. Используемые переменные 15](#_Toc104028397)

[3.2. Псевдокод 15](#_Toc104028398)

[3.3. Блок-схема 17](#_Toc104028399)

[3.4. Листинг программной реализации 20](#_Toc104028400)

[3.5. Результаты 21](#_Toc104028401)

[3.6. Тесты 22](#_Toc104028402)

[Задача №4 23](#_Toc104028403)

[1. Постановка задачи 23](#_Toc104028404)

[2. Структурные требования 23](#_Toc104028405)

[3. Описание задачи 23](#_Toc104028406)

[3.1. Используемые переменные 23](#_Toc104028407)

[3.2. Псевдокод 23](#_Toc104028408)

[3.3. Блок-схема 24](#_Toc104028409)

[3.4. Листинг программной реализации 26](#_Toc104028410)

[3.5. Результаты 27](#_Toc104028411)

[Задача №5 28](#_Toc104028412)

[1. Постановка задачи 28](#_Toc104028413)

[2. Структурные требования 28](#_Toc104028414)

[3. Описание задачи 28](#_Toc104028415)

[3.1. Используемые переменные 28](#_Toc104028416)

[3.2. Псевдокод 28](#_Toc104028417)

[3.3 Блок-схема 29](#_Toc104028418)

[3.4 Листинг программной реализации 31](#_Toc104028419)

[3.5. Результаты 32](#_Toc104028420)

[Задача №6 33](#_Toc104028421)

[1. Постановка задачи 33](#_Toc104028422)

[2. Структурные требования 33](#_Toc104028423)

[3. Описание задачи 33](#_Toc104028424)

[3.1. Используемые переменные 33](#_Toc104028425)

[3.2. Псевдокод 33](#_Toc104028426)

[3.3. Блок-схема 34](#_Toc104028427)

[3.4. Листинг программной реализации 38](#_Toc104028428)

[3.5. Результаты 40](#_Toc104028429)

[СПИСОК ЛИТЕРАТУРЫ 41](#_Toc104028430)

# Задача №1

## Постановка задачи

Транспонировать заданную матрицу А(N,N)

## Структурные требования

## Описание задачи

### Используемые переменные

n – кол-во строк и столбцов;

i – строки;

j – столбцы;

A[n][n] – двумерный массив.

### Псевдокод

Начало

Ввод: n

Если вывод: ошибка

Иначе для

Для

Вывод: A[i][j]

Для

Для

Вывод: A[i][j]

Конец

### Блок-схема

Начало

Ввод: n

Вывод: A[i][j]

Вывод: A[i][j]

вывод: ошибка

Конец

### Листинг программной реализации

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

srand(time(NULL));

int n;

printf("Введите n \n");

scanf("%i", &n);

int A[n][n];

if (n < 1) printf("Ошибка\n");

else

{

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

{

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

{

A[i][j] = ((RAND\_MAX) / rand());

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

}

printf("\n");

}

printf("\n");

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

{

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

{

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

}

printf("\n");

}

}

return 0;

}

### Результаты

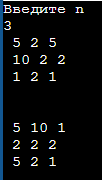


Рисунок – 1



Рисунок - 2

# Задача №2

## Постановка задачи

Найти сумму двух матриц А(N,M) и В(N,M)

## Структурные требования

## Описание задачи

### Используемые переменные

n – строки;

m – столбцы;

i – строки;

j – столбцы;

A[n][m], B[n][m], C[n][m] – двумерные массивы.

### Псевдокод

Начало

Ввод: n, m

Если Вывод: Ошибка

Иначе для

Для

Вывод: A[i][j]

Для

Для

Вывод: B[i][j]

Для

Для

Вывод: C[i][j]

Конец

### Блок-схема

Начало

Ввод: n, m

Вывод: A[i][j]

Вывод: B[i][j]

Вывод: C[i][j]

Конец

Вывод: Ошибка

### Листинг программной реализации

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

srand(time(NULL));

int n, m;

printf("Введите n и m \n");

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

int A[n][m], B[n][m], C[n][m];

if (n < 1 || m < 1 || n == m) printf("Ошибка\n");

else

{

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

{

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

{

A[i][j] = ((RAND\_MAX) / rand());

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

}

printf("\n");

}

printf("\n");

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

{

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

{

B[i][j] = ((RAND\_MAX) / rand());

printf(" %i", B[i][j]);

}

printf("\n");

}

printf("\n");

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

{

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

{

C[i][j] = A[i][j] + B[i][j];

printf(" %i", C[i][j]);

}

printf("\n");

}

}

return 0;

}

### Результаты

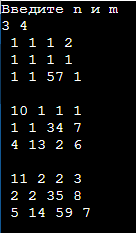


Рисунок – 3



Рисунок – 4

### Тесты

|  |  |
| --- | --- |
| Ручной | Программный |
| 11 | 11 |
| 2 | 2 |
| 3 | 2 |
| 22 | 3 |
| 35 | 2 |
| 8 | 2 |
| 35 | 35 |
| 8 | 8 |
| 5 | 5 |
| 14 | 14 |
| 59 | 59 |
| 7 | 7 |

# Задача №3

## Постановка задачи

Найти произведение двух матриц А(N,M) и В(M,K)

## Структурные требования

## Описание задачи

### Используемые переменные

n – кол-во строк для А и С;

m – кол-во столбцов для A, кол-во строк для B;

k – кол-во столбцов для B и С;

i – строки;

j – столбцы;

A[n][m], B[m][k], C[n][k] – двумерные массивы.

### Псевдокод

Начало

Ввод: n, m и k

Если Вывод: Ошибка

Иначе для

Для

Вывод: A[i][j]

Для

Для

Вывод: B[i][j]

Для

Для

Для

Вывод: C[i][j]

Конец

### Блок-схема

Начало

Ввод: n, m, k

Вывод: A[i][j]

Вывод: B[i][j]

Вывод: C[i][j]

Вывод: Ошибка

Конец

### Листинг программной реализации

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

srand(time(NULL));

int n, m, k;

printf("Введите n, m и k \n");

scanf("%i %i %i", &n, &m, &k);

int A[n][m], B[m][k], C[n][k];

if ((n < 1) || (m < 1) || (k < 1)) printf("Ошибка\n");

else

{

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

{

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

{

A[i][j] = ((RAND\_MAX) / rand());

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

}

printf("\n");

}

printf("\n");

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

{

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

{

B[i][j] = ((RAND\_MAX) / rand());

printf(" %i", B[i][j]);

}

printf("\n");

}

printf("\n");

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

{

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

{

C[i][j] = 0;

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

{

C[i][j] += A[i][f] \* B[f][j];

}

printf(" %i", C[i][j]);

}

printf("\n");

}

}

return 0;

}

### Результаты

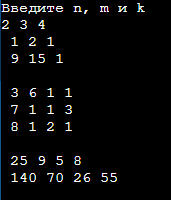


Рисунок – 5

### Тесты

|  |  |
| --- | --- |
| Ручной | Программный |
|  |  |

# Задача №4

## Постановка задачи

Найти произведение матрицы А(N,M) на вектор В(N)

## Структурные требования

## Описание задачи

### Используемые переменные

n – строки;

m – столбцы;

i – строки;

j – столбцы;

B(N) – целочисленный вектор состоящий из N элементов.

A[n][m], C[n][m] – двумерные массивы.

### Псевдокод

Начало

Ввод: n, m

Если Вывод: Ошибка

Иначе для

Для

Вывод: A[i][j]

Для

Вывод: B[i]

Для

Для

Вывод: C[i][j]

Конец

### Блок-схема

Начало

Ввод: n, m

Вывод: A[i][j]

Вывод: B[i]

Вывод: C[i][j]

Конец

Вывод: Ошибка

### Листинг программной реализации

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

srand(time(NULL));

int\* B, n, m;

printf("Введите n, m \n");

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

int A[n][m], C[n][m];

if ((n < 1) || (m < 1) || (n == m)) printf("Ошибка\n");

else

{

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

{

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

{

A[i][j] = ((RAND\_MAX) / rand());

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

}

printf("\n");

}

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

{

B[i] = ((RAND\_MAX) / rand());

printf("%i ", B[i]);

}

printf("\n");

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

{

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

{

C[i][j] = A[i][j] \* B[i];

printf(" %i", C[i][j]);

}

printf("\n");

}

}

return 0;

}

### Результаты

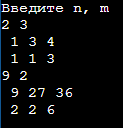


Рисунок – 6

# Задача №5

## Постановка задачи

В матрице А(N,M) найти номер строки, в которой сумма элементов наибольшая.

## Структурные требования

## Описание задачи

### 3.1. Используемые переменные

b – наибольшая сумма элементов строки;

d – номер строки, где находится b;

n – кол-во строк;

m – кол-во столбцов;

i – строки;

j – столбцы;

s – переменная, для поиска наибольшей суммы элементов строки;

A[n][m] – двумерный массив.

### 3.2. Псевдокод

Начало

Ввод: n, m

b = 1

Если Вывод: Ошибка

Иначе для

Для

Вывод: A[i][j]

Если

Вывод: d, b

Конец

### 3.3 Блок-схема

Начало

Ввод: n, m

Вывод: A[i][j]

b = 1

Вывод: d, b

Вывод: Ошибка

Конец

нет

да

### 3.4 Листинг программной реализации

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

srand(time(NULL));

int b, d, n, m, s;

printf("Введите n, m \n");

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

int A[n][m], C[n][m];

b = 1;

if ((n < 1) || (m < 1) || (n == m)) printf("Ошибка\n");

else

{

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

{

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

{

A[i][j] = ((RAND\_MAX) / rand());

s += A[i][j];

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

}

if (s > b)

{

b = s;

d = i + 1;

}

s = 0;

printf("\n");

}

printf("\nВ %i строке находится наибольшая сумма элементов = %i", d, b);

}

return 0;

}

### 3.5. Результаты

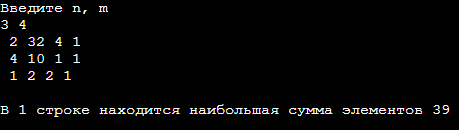


Рисунок – 7

# Задача №6

## Постановка задачи

В матрице А(N,M) переставить местами строки содержащие первый максимальный элемент и последний минимальный

## Структурные требования

## Описание задачи

### Используемые переменные

n – кол-во строк;

m – кол-во столбцов;

i – строки;

j – столбцы;

x – переменная, которая используется в методе 3 стаканов;

in – строка, в которой находится минимальный элемент;

ix – строка, в которой находится максимальный элемент;

mn – минимальный элемент;

mx – максимальный элемент;

A[n][m] – двумерный массивы.

### Псевдокод

Начало

Ввод: n, m

Если Вывод: Ошибка

Иначе для

Для

Вывод: A[i][j]

Для

Для

Если

Если

Для

Для

Для

Вывод: A[i][j]

Конец

### Блок-схема

Начало

Ввод: n, m

Вывод: A[i][j]

Вывод: A[i][j]

Вывод: Ошибка

Конец

### Листинг программной реализации

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

srand(time(NULL));

int n, m, x, in, ix, mn, mx;

printf("Введите n, m \n");

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

int A[n][m];

if ((n < 1) || (m < 1) || (n == m)) printf("Ошибка\n");

else

{

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

{

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

{

A[i][j] = rand() % 10;

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

mn = A[0][0];

mx = A[0][0];

in = 0;

ix = 0;

}

printf("\n");

}

printf("\n");

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

{

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

{

if (A[i][j] >= mx)

{

mx = A[i][j];

ix = i;

}

if (A[i][j] <= mn)

{

mn = A[i][j];

in = i;

}

}

}

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

{

x = A[in][j];

A[in][j] = A[ix][j];

A[ix][j] = x;

}

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

{

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

{

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

}

printf("\n");

}

}

return 0;

}

### Результаты

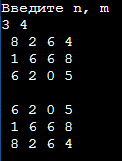


Рисунок – 8

# СПИСОК ЛИТЕРАТУРЫ

1. Конова Е.А., Поллак Г.А. Алгоритмы и программы. Язык С++: Учебное

пособие. - 2-е изд., стер. - СПб.: Издательство "Лань", 2017. - 384 с.

2. Седжвик Роберт. Алгоритмы на С++.: Пер. с англ. - М.: ООО "И.Д. Вильямс",

2011. - 1056 с.

3. Лафоре Р. Объектно-ориентированное программирование в С++. Классика

Computer Science. 4-е изд. - СПб.: Питер, 2015. - 928 с.

4. Орлов С.А. Теория и практика языков программирования: Учебник для вузов.

Стандарт 3-го поколения. - СПб.: Питер, 2014. - 688 с.

5. Павловская Т.А. С/С++. Процедурное и объектно-ориентированное

программирование: Учебник для вузов. Стандарт 3-го поколения. - СПб.:

Питер, 2015. - 496 с.

6. Павловская Т.А. С#. Программирование на языке высокого уровня: Учебник

для вузов. - СПб.: Питер, 2014. - 432 с.

7. Плаксин М.А. Тестирование и отладка программ - для профессионалов

будущих и настоящих. - М.: БИНОМ. Лаборатория знаний, 2013. - 167 с.