Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ О ПРАКТИЧЕСКОЙ** **РАБОТЕ №2**

Динамические массивы (многомерные)

Тема

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Преподаватель |  |  |  |  |  | Черниговский А.С |
|  |  |  |  | подпись, дата |  | инициалы, фамилия |
| Студент |  | КИ19-16/2б |  |  |  | Шпаков С.Е. |
|  |  | номер группы, зачетной книжки |  | подпись, дата |  | инициалы, фамилия |

Красноярск 2020

# СОДЕРЖАНИЕ

[1 Цель и задачи 3](#_Toc27678199)

[1.1 Цель 3](#_Toc27678200)

[1.2 Задачи 3](#_Toc27678201)

[2 Описание варианта задания 3](#_Toc27678202)

[2.1 Основное задание 3](#_Toc27678203)

[2.2 Дополнительное задание 4](#_Toc27678204)

[3 Ход выполнения 4](#_Toc27678205)

[4 Выводы 11](#_Toc27678206)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 12](#_Toc27678207)

# 1 Цель и задачи

## 1.1 Цель

Написать программу, выполняющую свои функции и выполняющую задачи ниже.

## 1.2 Задачи

На оценку 3 балла:

1. отформатировать исходный код программы согласно требованиям стандарта кодирования;
2. размерность массива должна определяться пользователем;
3. организовать повтор программы по желанию пользователя.

На оценку 4 балла:

1. выполнить все предыдущие пункты;
2. реализовать проверку входных параметров;
3. реализовать меню пользователя, состоящее как минимум из 4-х пунктов (ввод данных, обработка данных, вывод результата на экран, выход);
4. для корректной работы меню организовать промежуточное хранение результата.

На оценку 5 баллов:

1. выполнить все предыдущие пункты;
2. реализовать меню пользователя, состоящее минимум из 4-х пунктов (ввод элементов массива, обработка массива, вывод массива на экран, выход);
3. предусмотреть заполнение массива случайными величинами, либо значениями, введенными пользователем с клавиатуры, если это возможно..

# 2 Описание варианта задания

Вариант №24. Дана матрица, содержащая N строк и M столбцов. Написать программу позволяющую повернуть матрицу на 90, 180, 270 градусов.

# 3 Ход выполнения

Необходимо создать и заполнить двумерный массив, размерность которого определяется пользователем. Заполнение массива на выбор случайными числами или с клавиатуры.

Обработка матрицы включает в себя запрос у пользователя угла, кратного 90 градусам.

В меню пользователя необходимо возможность повторения действий и выход из программы. Программа реализована ниже (листинг 1).

Листинг 1 – Реализованная программа с комментариями

#include <stdio.h>

#include <malloc.h>

#include <time.h>

int\*\* input(int\* vsize, int\* hsize)

{

int n = 0;

int m = 0;

char check = '\0';

int const range = 40;

// Asking about the size

printf("\nHow many lines should matrix have?\n");

while (scanf("%d%c", &n, &check, 1) != 2 || check != '\n' || n <= 0)

{

printf("Please enter valid and positive number");

while (getchar() != '\n')

;

}

check = '\0';

printf("\nHow many rows should matrix have?\n");

while (scanf("%d%c", &m, &check, 1) != 2 || check != '\n' || n <= 0)

{

printf("Please enter valid and positive number");

while (getchar() != '\n')

;

}

// Manual or random input

int fill = 0;

check = '\0';

printf("\nHow do you want to fill the array?\n1. Random\n2. Manual\n");

while ((scanf("%d%c", &fill, &check, 1) != 2 || check != '\n') || (fill != 1 && fill != 2))

{

printf("No such option\n");

while (getchar() != '\n')

;

}

// "random"

srand(time(NULL));

Продолжение Листинга 1.

// Memory allocation

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

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

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

int num = 0;

// Filling with numbers manually or randomly depending on the answer above

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

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

{

if (fill == 2)

{

num = 0;

check = '\0';

printf("(%d;%d) = ", i, j);

while (scanf("%d%c", &num, &check, 1) != 2 || check != '\n') {

printf("\nPlease enter valid number\n");

while (getchar() != '\n');

}

arr[i][j] = num;

}

else if (fill == 1)

arr[i][j] = rand() % range;

};

\*vsize = n;

\*hsize = m;

return arr;

}

// Func to free matrix

void arrfree(int \*\*\*arr, int n)

{

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

{

int\*\* dummy = (int \*\*)arr[i];

free(dummy);

}

free(\*arr);

}

int main() {

int vsize=0, hsize=0;

int\*\* source = NULL;

int choice;

// Infinite loop of choices

while (1)

{

printf("\nWhatcha want?\n1. Enter new info\n2. "

"Rotate matrix\n3. Output matrix\n4. Exit\n");

char checker = '\0';

while (scanf("%d%c", &choice, &checker, 1) != 2 || checker != '\n') {

printf("\nPlease enter valid number\n");

while (getchar() != '\n');

}

switch (choice)

{

Продолжение Листинга 1.

// New matrix. Frees previous one if it exists

case 1:

if (source != NULL)

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

{

int\* dummy = source[i];

free(dummy);

}

free(source);

source = input(&vsize, &hsize);

break;

// Proceed matrix. No func cause I forgot

case 2:

if (source != NULL)

{

int number = 0;

char check = '\0';

printf("\nRotate by how many degrees counterclock-wise?\n");

while (scanf("%d%c", &number, &check, 1) != 2 || check != '\n' || number % 90 != 0) {

printf("\nPlease enter a number multiple of 90\n");

while (getchar() != '\n');

}

// Checking for the angle and rotating

// Nothing much to say. Just refilling

switch ((number % 360)) {

case 0:

break;

case 90: {

int \*\*mirror = (int \*\*) malloc(sizeof(int \*) \* vsize);

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

mirror[i] = (int \*) malloc(sizeof(int) \* hsize);

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

mirror[i][j] = source[i][j];

}

arrfree(&source, vsize);

source = (int \*\*) malloc(sizeof(int \*) \* hsize);

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

source[i] = (int \*) malloc(sizeof(int) \* vsize);

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

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

source[i][j] = mirror[j][hsize - 1 - i];

arrfree(&mirror, vsize);

int k = hsize;

hsize = vsize;

vsize = k;

break;

}

case 180: {

int \*\*mirror = (int \*\*) malloc(sizeof(int \*) \* vsize);

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

mirror[i] = (int \*) malloc(sizeof(int) \* hsize);

Продолжение Листинга 1.

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

mirror[i][j] = (int) source[i][j];

}

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

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

source[i][j] = (int) mirror[vsize - 1 - i][hsize - 1 - j];

}

arrfree(&mirror, vsize);

break;

}

case 270: {

int \*\*mirror = (int \*\*) malloc(sizeof(int \*) \* vsize);

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

mirror[i] = (int \*) malloc(sizeof(int) \* hsize);

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

mirror[i][j] = source[i][j];

}

arrfree(&source, vsize);

source = (int \*\*) malloc(sizeof(int \*) \* hsize);

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

source[i] = (int \*) malloc(sizeof(int) \* vsize);

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

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

source[i][j] = mirror[vsize-1-j][i];

arrfree(&mirror, vsize);

int k = hsize;

hsize = vsize;

vsize = k;

break;

}

default:

break;

}

break;

}

// Output on screen.

case 3:

if (source != NULL)

{

printf("\n");

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

{

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

printf("%-3d ", source[i][j]);

printf("\n");

}

}

else

printf("\nIt is not yet proceeded\n");

break;

case 4:

return 0;

Продолжение Листинга 1.

default:

printf("There is no such option :c");

break;

}

}

}

}

}

}

# 4 Выводы

Задача реализована на языке Си. Программа поворачивает матрицу на количество градусов, кратное 90, согласно условию задачи.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. СТО 4.2-07-2014 Система менеджмента качества Общие требования к построению, изложению и оформлению документов учебной деятельности. – Взамен СТО 4.2–07–2012 ; введ. 09.01.2014, - Красноярск : ИПК СФУ, 2014 – 60 с.