**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**

**МЕХАНИКИ И ОПТИКИ”**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ**

**по Лабораторной работе № 18-22**

**«Программирование задач с использованием двумерных массивов. Ввод, вывод, упорядочивание. Вычисление характеристик массива»**

Специальность 09.02.03 «Программирование в компьютерных системах»

Дисциплина «Основы программирования»

|  |  |
| --- | --- |
| Преподаватель:  Данилина Татьяна Викторовна  «\_21.06\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2018г.  Оценка \_\_\_\_\_\_\_ОТЛ\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2235  Матвеенко Дмитрий Владимирович |

Санкт-Петербург

2017/2018

**Цель**

Получение практических навыков в использовании двумерных массивов - матриц.

**Задание**

1. Сумма/произведение/количество в каждой строке по условию.

2. Найти минимум, максимум каждой строки; поменять их местами; вывести измененный массив на экран.

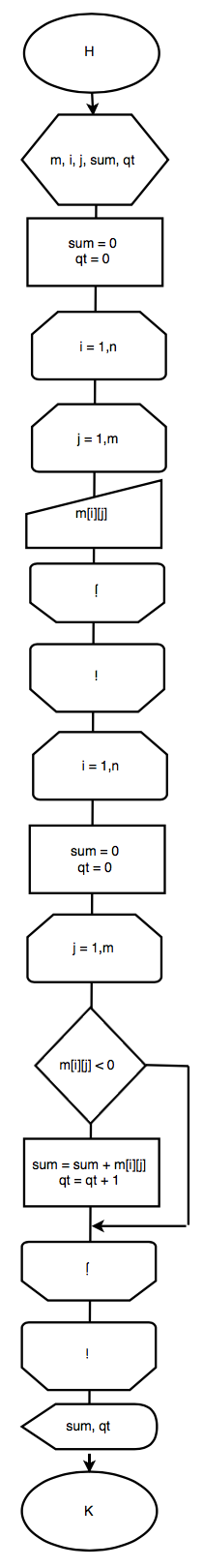
3. Вычислить и запомнить суммы и число отрицательных элементов каждой строки матрицы. Результат отпечатать в виде 2-х столбцов.

4. Максимальное из чисел, встречающихся в заданной матрице более одного раза.

5. Характеристикой строки целочисленной матрицы назовем сумму её положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.

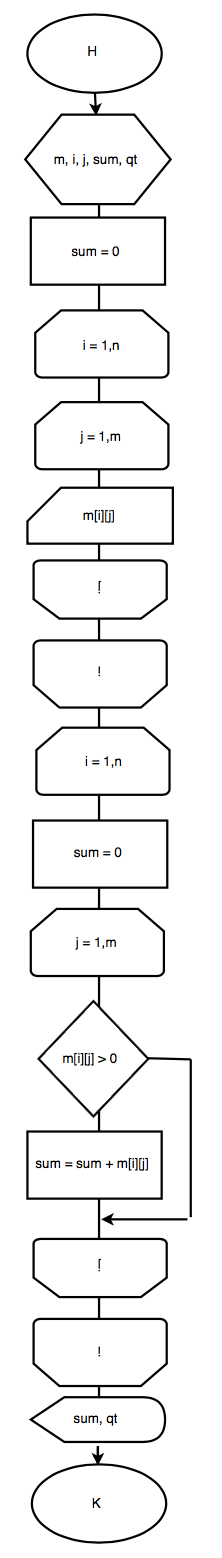
**АЛГОРИТМ**

Алгоритм решения задачи номер 3 представлен ниже (см. Рисунок 1)



*Рисунок 1*

Алгоритм решения задачи номер 2 представлен ниже (см. Рисунок 2)

**

*Рисунок 2*

**ТЕКСТ ПРОГРАММЫ**

Текст программы для задачи номер 1, представлен ниже.

#include <stdio.h>

int main() {

int m[2][3];

int i, j;

int sum = 0;

puts("Введите массив ниже:");

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

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

//printf("a[%d][%d] = ", i, j);

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

}

}

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

sum = 0;

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

if (m[i][j] > 0) {

sum += m[i][j];

}

}

printf("Ответ: %d\n", sum);

}

return 0;

}

Текст программы для задачи номер 2, представлен ниже.

#include <stdio.h>

int main() {

int m[2][3];

int i, j;

int min = 0;

int max = 0;

int swapOne = 0;

int swapTwo = 0;

puts("Введите массив ниже:");

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

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

//printf("a[%d][%d] = ", i, j);

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

}

}

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

min = m[i][0];

max = m[i][0];

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

if (m[i][j] >= max) {

max = m[i][j];

swapOne = j;

}

if (m[i][j] < min) {

min = m[i][j];

swapTwo = j;

}

}

m[i][swapTwo] = max;

m[i][swapOne] = min;

}

printf("Ответ:\n");

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

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

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

}

printf("\n");

}

return 0;

}

Текст программы для задачи номер 3, представлен ниже.

#include <stdio.h>

int main() {

int m[2][3];

int i, j;

int sum = 0;

int qt = 0;

puts("Введите массив ниже:");

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

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

//printf("a[%d][%d] = ", i, j);

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

}

}

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

sum = 0;

qt = 0;

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

if (m[i][j] < 0) {

sum += m[i][j];

qt ++;

}

}

printf("Сумма: %d Количество: %d\n", sum, qt);

}

return 0;

}

Текст программы для задачи номер 4, представлен ниже.

#include <stdio.h>

int main() {

int m[3][3];

int i, j;

int stop = 0;

int repeat = -10000, repeat1 = -10000, repeat2 = 0;

puts("Введите массив ниже:");

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

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

//printf("a[%d][%d] = ", i, j);

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

}

}

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

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

repeat2 = m[i][j];

m[i][j] = -10000;

for (int t = 0; t < 3; t++) {

for (int p = 0; p < 3; p++) {

if (repeat2 == m[t][p]) {

repeat = m[t][p];

stop++;

}

}

if (repeat1 > repeat) {

repeat = repeat1;

}

}

repeat1 = repeat;

m[i][j] = repeat2;

}

}

if (stop > 0) {

printf("Ответ: %i\n", repeat);

} else {

printf("Нет повторяющегося максимума\n");

}

return 0;

}

Текст программы для задачи номер 5, представлен ниже.

#include <stdio.h>

#define n 4

#define m 4

int main(){

int i, j, k;

int str[n];

int matr[n][m];

puts("Введите массив ниже:");

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

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

//printf("a[%d][%d] = ", i, j);

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

}

}

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

str[i] = 0;

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

if ((matr[i][j] > 0) && (matr[i][j] % 2 == 0)) {

str[i] += matr[i][j];

}

}

}

//Переставляем строки

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

k = i;

for (j = i + 1; j < m; j++) {

if (str[j] < str[k]) {

k = j;

}

}

if (k != i) {

//Меняю минимальную строку на первую

int \*a = &str[k];

int \*b = &str[i];

int t = \*a;

\*a = \*b;

\*b = t;

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

int \*a = &matr[i][j];

int \*b = &matr[k][j];

int t = \*a;

\*a = \*b;

\*b = t;

}

}

}

printf("Ответ:\n");

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

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

printf("% 2d ", matr[i][j]);

}

putchar('\n');

}

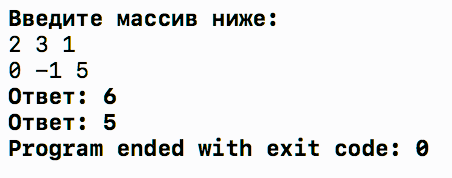
return 0;

}

}

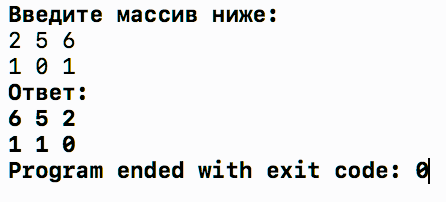
**ПРОТОКОЛ**

Протокол программы для задачи номер 1, представлен ниже (см. Рисунок 3)



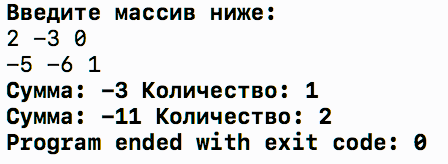
*Рисунок**3*

Протокол программы для задачи номер 2, представлен ниже (см. Рисунок 4)



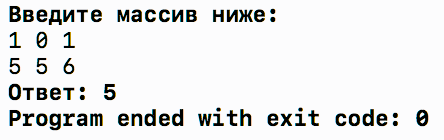
*Рисунок**4*

Протокол программы для задачи номер 3, представлен ниже (см. Рисунок 5)

****

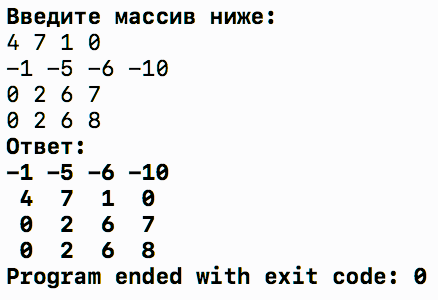
*Рисунок**5*

Протокол программы для задачи номер 4, представлен ниже (см. Рисунок 6)

****

*Рисунок**6*

Протокол программы для задачи номер 5, представлен ниже (см. Рисунок 7)



*Рисунок**7*

**ВЫВОД**

1. Массив – это фиксированное количество упорядоченных компонент одного типа, снабжённых индексами.

2. Чтобы описать массив, надо сообщить компилятору: сколько в нем элементов, какого типа эти элементы, как они нумеруются. Элементы массива нумеруются с нуля.

3. Доступ к элементу массива осуществляется путем указания индекса (номера) элемента. Доступ к элементам массива можно осуществлять при помощи указателя. В инструкции объявления массива удобно использовать именованную константу, объявленную в директиве #define.

4. Для ввода, вывода и обработки массивов удобно использовать инструкции циклов (for, while).

5. Типичной ошибкой при использовании массивов является обращение к несуществующему элементу, то есть выход индекса за допустимое значение.