**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное автономное образовательное**

**учреждение высшего образования**

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(РУТ(МИИТ)**

Кафедра «Вычислительные системы, сети и информационная безопасность»

**Отчет По Дисциплине**

**«Алгоритмизация и программирование»**

**Лабораторная работа №7**

*Направление:* 10.03.01*Информационная безопасность*

*Профиль:**Безопасность компьютерных систем*

Выполнил:  
студент группы УИБ-112

Орлов Андрей Витальевич

Проверил:

Старший преподаватель Никольская Мария Николаевна

(должность, ФИО)

Старший преподаватель Цыганова Наталия Алексеевна

(должность, ФИО)

Москва 2021 г.

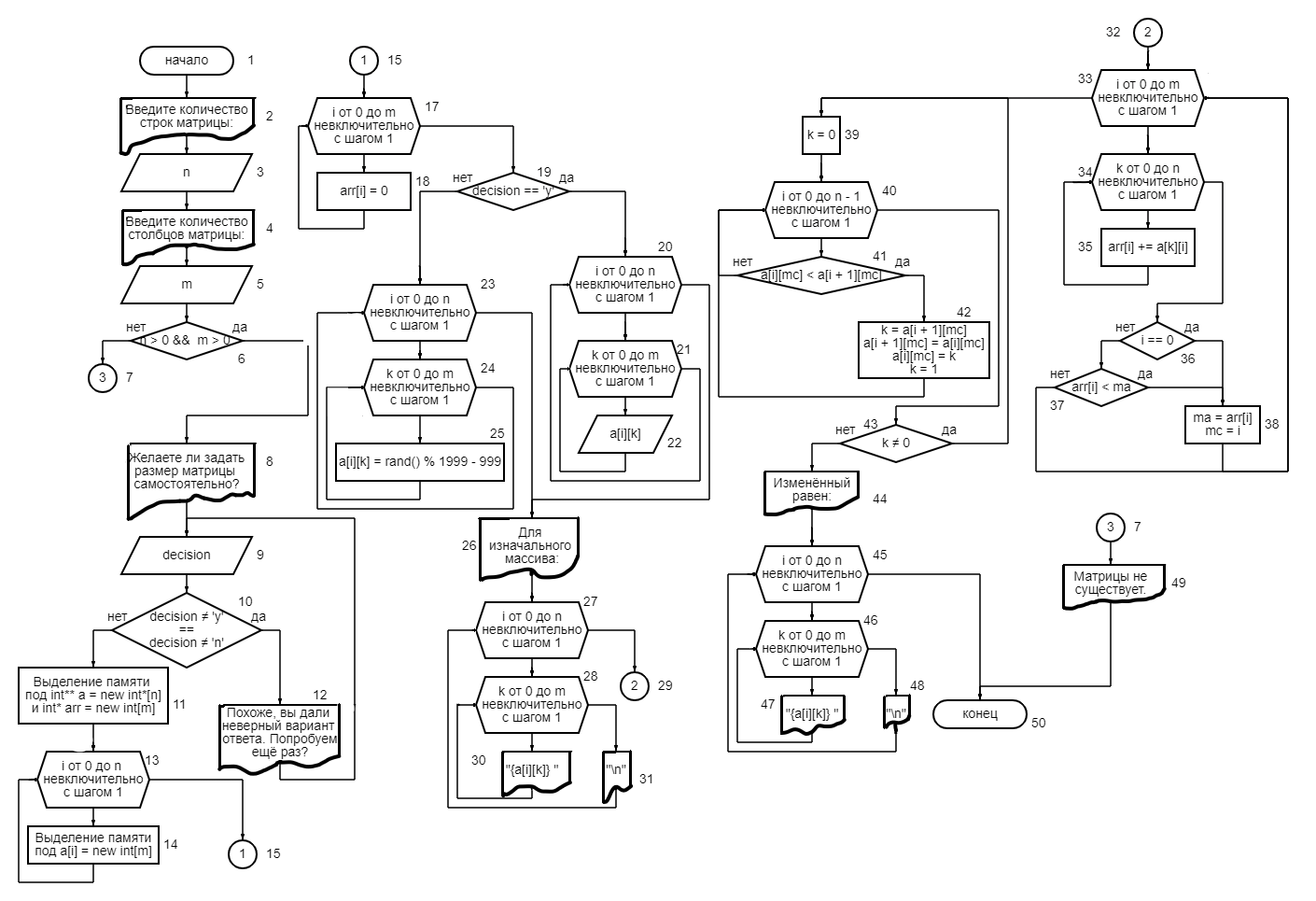
**Задание №7**

Задание: Задана матрица целых чисел. Выполнить сортировку элементов по убыванию в столбце с минимальной суммой элементов.

1. Таблица имён:

|  |  |  |
| --- | --- | --- |
| Исходные данные | | |
| n | целочисленный | Количество строк в матрице |
| m | целочисленный | Количество столбцов в матрице |
| a[n][m] | целочисленный | Исходная матрица |
| decision | символьный | Определитель выбора ввода матрицы |
| Рабочие переменные | | |
| arr[m] | целочисленный | Массив сумм столбцов матрицы |
| a[i][k] | целочисленный | Элемент матрицы i строки j столбца |
| ma | целочисленный | Минимальная сумма в столбце |
| mc | целочисленный | Индекс столбца с минимальной суммой |
| Результат | | |
| a[n][m] | целочисленный | Отсортированная исходная матрица |

1. Блок-схема:



1. Отладочный пример:

Отладочный пример №1:  
1. Начало  
2. Вывод «Введите количество строк матрицы:»  
3. Ввод n = 0  
4. Вывод «Введите количество столбцов матрицы»  
5. Ввод m = 0  
6. 0 > 0 && 0 > 0, нет, переход в блок 7.  
7. Ссылка на блок 49  
49. Вывод «Матрицы не существует. »   
50. Конец

Отладочный пример №2:  
1. Начало  
2. Вывод «Введите количестве строк в матрице »  
3. Ввод n = 3  
4. Вывод «Введите количестве столбцов в матрице »  
5. Ввод m = 3  
6. 3 > 0 && 3 > 0, да, переход в блок 8  
8. Вывод «Желаете ли вы задать размер матрицы соответственно?»  
9. Ввод decision = “Error”  
10. “Error” != ‘y’ == “Error” != ‘n’, да, идём в блок 12  
12. Вывод «Похоже, вы дали неверный вариант ответа. Попробуем ещё раз?»  
9. Ввод 'y'  
10. “y” != ‘y’ == “y” != ‘n’, нет, идём в блок 11  
11. Выделение памяти массиву массивов а размерности 3.  
Выделение памяти массиву arr размерности 3.  
13. Вход в цикл: Переменная i = 0  
14. Выделение памяти элементу массива массивов a[0] размерности 3.  
13. Вход в цикл: Переменная i = 1  
14. Выделение памяти элементу массива массивов a[1] размерности 3.  
13. Вход в цикл: Переменная i = 2  
14. Выделение памяти элементу массива массивов a[2] размерности 3.  
13. Вход в цикл: Переменная i = 3. Граница достигнута, переход в блок 15  
15. Ссылка на блок 17  
17. Вход в цикл: Переменная i = 0.  
18. Переприсваивание переменных arr[0] = 0.  
17. Вход в цикл: Переменная i = 1.  
18. Переприсваивание переменных arr[1] = 0.  
17. Вход в цикл: Переменная i = 2.  
18. Переприсваивание переменных arr[2] = 0.  
17. Вход в цикл. Переменная i = 3. Граница достигнута, переход к блоку 19  
19. 'y' == 'y', да, идём в блок 20  
20. Вход в цикл: Переменная i = 0  
21. Вход в цикл: Переменная k = 0  
22. Ввод a[0][0] = 2  
21. Вход в цикл: Переменная k = 1  
22. Ввод a[0][1] = 7  
21. Вход в цикл: Переменная k = 2. Граница достигнута, переход к блоку 20  
22. Ввод a[0][2] = 3  
20. Вход в цикл: Переменная i = 1  
21. Вход в цикл: Переменная k = 0  
22. Ввод a[1][0] = 5  
21. Вход в цикл: Переменная k = 1  
22. Ввод a[1][1] = 1  
21. Вход в цикл: Переменная k = 2. Граница достигнута, переход к блоку 20  
22. Ввод a[1][2] = 6  
20. Вход в цикл: Переменная i = 2  
21. Вход в цикл: Переменная k = 0  
22. Ввод a[2][0] = 8  
21. Вход в цикл: Переменная k = 1  
22. Ввод a[2][1] = 4  
21. Вход в цикл: Переменная k = 2. Граница достигнута, переход к блоку 20  
22. Ввод a[2][2] = 9  
20. Вход в цикл: Переменная i = 3. Граница достигнута, переход к блоку 26  
26. Вывод «Для изначального массива: »  
27. Вход в цикл: Переменная i = 0  
28. Вход в цикл: Переменная k = 0  
30. Вывод «2 »  
28. Вход в цикл: Переменная k = 1  
30. Вывод «7»  
28. Вход в цикл: Переменная k = 2  
30. Вывод «3 »  
28. Вход в цикл: Переменная k = 3. Граница достигнута, переход к блоку 27  
27. Вход в цикл: Переменная i = 1  
28. Вход в цикл: Переменная k = 0  
30. Вывод «5 »  
28. Вход в цикл: Переменная k = 1  
30. Вывод «1 »  
28. Вход в цикл: Переменная k = 2  
30. Вывод «6 »  
28. Вход в цикл: Переменная k = 3. Граница достигнута, переход к блоку 27  
27. Вход в цикл: Переменная i = 2  
28. Вход в цикл: Переменная k = 0  
30. Вывод «8 »  
28. Вход в цикл: Переменная k = 1  
30. Вывод «4 »  
28. Вход в цикл: Переменная k = 2  
30. Вывод «9 »  
28. Вход в цикл: Переменная k = 3. Граница достигнута, переход к блоку 27  
27. Вход в цикл: Переменная i = 3. Граница достигнута, переход к блоку 29  
29. Ссылка на блок 33  
33. Вход в цикл: Переменная i = 0  
34. Вход в цикл: Переменная k = 0  
35. arr[0] += 2 = 0 + 2 = 2  
34. Вход в цикл: Переменная k = 1  
35. arr[0] += 5 = 2 + 5 = 7  
34. Вход в цикл: Переменная k = 2  
35. arr[0] += 8 = 7 + 8 = 15  
34. Вход в цикл: Переменная k = 3. Граница достигнута, переход в блок 36  
36. 0 == 0, да, переход в блок 38  
38. Переприсваивание переменных ma = 15, mc = 0  
33. Вход в цикл. Переменная i = 1  
34. Вход в цикл: Переменная k = 0  
35. arr[1] += 7 = 0 + 7 = 7  
34. Вход в цикл: Переменная k = 1  
35. arr[1] += 1 = 7 + 1 = 8  
34. Вход в цикл: Переменная k = 2  
35. arr[1] += 4 = 8 + 4 = 12  
34. Вход в цикл: Переменная k = 3. Граница достигнута, переход в блок 36  
36. 1 == 0, нет, переход в блок 37  
37. 12 < 15, да, переход в блок 38  
38. Переприсваивание переменных ma = 12, mc = 1  
33. Вход в цикл. Переменная i = 2  
34. Вход в цикл: Переменная k = 0  
35. arr[2] += 3 = 0 + 3 = 3  
34. Вход в цикл: Переменная k = 1  
35. arr[2] += 6 = 3 + 6 = 9  
34. Вход в цикл: Переменная k = 2  
35. arr[2] += 9 = 9 + 9 = 18  
34. Вход в цикл: Переменная k = 3. Граница достигнута, переход в блок 36  
36. 2 == 0, нет, переход в блок 37  
37. 18 < 12, нет, переход в блок 33  
33. Вход в цикл. Переменная i = 2. Граница достигнута, переход в блок 39  
39. Переприсваивание переменных k = 0  
40. Вход в цикл: Переменная i = 0  
41. 7 < 1, нет, идём в блок 40  
40. Вход в цикл: Переменная i = 1  
41. 1 < 4, да, идём в блок 42  
42. Переприсваивание переменных  
k = 4; a[2][1] = 1; a[1][1] = 4; k = 1  
40. Вход в цикл: Переменная i = 2. Граница достигнута, переход в 43  
43. 1 != 0, нет, идём в блок 39  
39. Переприсваивание переменных k = 0  
40. Вход в цикл: Переменная i = 0  
41. 7 < 4, нет, идём в блок 40  
40. Вход в цикл: Переменная i = 1  
41. 4 < 1, нет, идём в блок 40  
40. Вход в цикл: Переменная i = 2. Граница достигнута, переход в 43  
43. 0 != 0, нет, переход в блок 44  
44. Вывод «Изменённый массив равен: »  
45. Вход в цикл: Переменная i = 0  
46. Вход в цикл: Переменная k = 0  
47. Вывод «2 »  
46. Вход в цикл: Переменная k = 1  
47. Вывод «5 »  
46. Вход в цикл: Переменная k = 2  
47. Вывод «8 »  
46. Вход в цикл: Переменная k = 3. Граница достигнута, переход в блок 48  
48. Перенос текста на новую строку  
45. Вход в цикл: Переменная i = 1  
46. Вход в цикл: Переменная k = 0  
47. Вывод «7 »  
46. Вход в цикл: Переменная k = 1  
47. Вывод «4 »  
46. Вход в цикл: Переменная k = 2  
47. Вывод «1 »  
46. Вход в цикл: Переменная k = 3. Граница достигнута, переход в блок 48  
48. Перенос текста на новую строку  
45. Вход в цикл: Переменная i = 2  
46. Вход в цикл: Переменная k = 0  
47. Вывод «3 »  
46. Вход в цикл: Переменная k = 1  
47. Вывод «6 »  
46. Вход в цикл: Переменная k = 2  
47. Вывод «9 »  
46. Вход в цикл: Переменная k = 3. Граница достигнута, переход в блок 48  
48. Перенос текста на новую строку  
45. Вход в цикл: Переменная i = 3. Граница достигнута, переход в блок 50  
50. Конец  
  
Отладочный пример №3

1. Начало  
2. Вывод «Введите количество строк матрицы»  
3. Ввод n = 2  
4. Вывод «Введите количество столбцов в матрице»  
5. Ввод m = 2  
6. 2 > 0 && 2 > 0, да, переход к блоку 8   
8. Вывод «Желаете ли задать размер матрицы самостоятельно7»  
9. Ввод decision = 'n'  
10. decision != y && decision != 'n', нет, переход в блок 11  
11. Выделение памяти под массив массивов a размерности 2  
Выделение памяти под массив сумм arr размерности 2  
13. Вход в цикл: Переменная i = 0  
14. Выделение памяти элементу массива массивов a[0] размерности 2  
13. Вход в цикл: Переменная i = 1  
14. Выделение памяти элементу массива массивов a[1] размерности 2  
13. Вход в цикл: Переменная i = 2. Граница достигнута, переход в блок 15  
15. Ссылка на блок 17  
17. Вход в цикл: Переменная i = 0  
18. Переприсваивание переменных arr[0] = 0  
17. Вход в цикл: Переменная i = 1  
18. Переприсваивание переменных arr[1] = 0  
17. Вход в цикл: Переменная i = 2. Граница достигнута, переход в блок 19.  
19. ‘n’ == ‘y’, нет, переход к блоку 23.  
23. Вход в цикл: Переменная i = 0  
24. Вход в цикл: Переменная k = 0  
25. Переприсваивание переменных a[0][0] = 283  
24. Вход в цикл: Переменная k = 1  
25. Переприсваивание переменных a[0][1] = 200  
24. Вход в цикл: Переменная k = 2. Граница достигнута, переход в блок 23  
23. Вход в цикл: Переменная i = 0  
24. Вход в цикл: Переменная k = 1  
25. Переприсваивание переменных a[1][0] = 115  
24. Вход в цикл: Переменная k = 1  
25. Переприсваивание переменных a[1][1] = 639  
24. Вход в цикл: Переменная k = 2. Граница достигнута, переход в блок 23  
23. Вход в цикл: Переменная i = 2. Граница достигнута, переход в блок 26  
26. Вывод «Для изначального массива »  
23. Вход в цикл: Переменная i = 0  
24. Вход в цикл: Переменная k = 0  
25. Вывод «283 »  
24. Вход в цикл: Переменная k = 1  
25. Вывод «200 »  
24. Вход в цикл: Переменная k = 2. Граница достигнута, переход в блок 31  
31. Перевод текста на новую строку  
23. Вход в цикл: Переменная i = 0  
24. Вход в цикл: Переменная k = 1  
25. Вывод «115»  
24. Вход в цикл: Переменная k = 1  
25. Вывод «639»  
24. Вход в цикл: Переменная k = 2. Граница достигнута, переход в блок 31  
31. Перевод текста на новую строку  
23. Вход в цикл: Переменная i = 2. Граница достигнута, переход в блок 29  
29. Ссылка на блок 33  
33. Вход в цикл: Переменная i = 0  
34. Вход в цикл: Переменная k = 0  
35. Переприсваивание переменных arr[0] += a[0][0] = 0 + 283 = 283  
34. Вход в цикл: Переменная k = 1  
35. Переприсваивание переменных arr[0] += a[0][1] = 283 + 115 = 398  
34. Вход в цикл: Переменная k = 2, граница достигнута, переход в блок 36  
36. 0 == 0, да, переход к блоку 38  
38 Переприсваивание переменных ma = 398, mc = 0  
33. Вход в цикл: Переменная i = 1  
34. Вход в цикл: Переменная k = 0  
35. Переприсваивание переменных arr[1] += a[1][0] = 0 + 200 = 200  
34. Вход в цикл: Переменная k = 1  
35. Переприсваивание переменных arr[1] += a[1][1] = 200 + 639 = 839  
34. Вход в цикл: Переменная k = 2, граница достигнута, переход в блок 36  
36. 1 == 0, нет, переход у блоку 37  
37. 839 < 398, нет, переход к блоку 33.  
33. Вход в цикл: Переменная i = 2. Граница достигнута, переход к блоку 39  
39. Переприсваивание переменных k = 0  
40. Вход в цикл: Переменная i = 0  
41. 283 < 115, нет, переход к блоку 40  
40. Вход в цикл: Переменная i = 1. Граница достигнута, переход к блоку 43  
43ю 0 != 0, нет, переход к блоку 44  
44. Вывод «Изменённый равен: »  
45. Вход в цикл: Переменная i = 0  
46. Вход в цикл: Переменная k = 0  
47. Вывод «283 »  
46. Вход в цикл: Переменная k = 1  
47. Вывод «200 »  
46. Вход в цикл: Переменная k = 2. Граница достигнута, переход в блок 48  
48. Перевод текста на новую строку  
45. Вход в цикл: Переменная i = 0  
46. Вход в цикл: Переменная k = 1  
47. Вывод «115»  
46. Вход в цикл: Переменная k = 1  
47. Вывод «639»  
46. Вход в цикл: Переменная k = 2. Граница достигнута, переход в блок 48  
45. Вход в цикл: Переменная i = 2. Граница достигнута, переход в блок 50  
50. Конец

1. Код программы:

#include <iostream>

#include <ctime>

using namespace std;

int main() {

srand(time(NULL)); // Для того, чтобы rand() генерировала постоянно новые числа

setlocale(LC\_ALL, "rus");

int n, m, ma, mc, i, k; char decision;

cout << "Введите количество строк матрицы: ";

cin >> n;

cout << "Введите количество столбцов матрицы: ";

cin >> m;

cout << "Желаете ли задать размер матрицы самостоятельно? (y/n)" << endl;

cin >> decision;

while (decision != 'y' && decision != 'n') {

cout << "Похоже, вы дали неверный вариант ответа. Попробуем ещё раз?\n";

cout << "Желаете ли задать элементы матрицы самостоятельно? (y/n)" << endl;

cin >> decision;

}

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

cout << "Матрицы не существует." << endl;

system("pause");

return 0;

}

int\*\* a = new int\* [n], \* arr = new int[m];

//Присваиваем каждому элементу массива размерность m

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

a[i] = new int[m];

}

//Присваиваем каждому элементу массива суммы значение 0

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

arr[i] = 0;

}

// Ввод массива пользователем

if (decision == 'y') {

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

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

cin >> a[i][k];

}

}

}

// Ввод массива компьютером

else {

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

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

a[i][k] = rand() % 1999 - 999;

}

}

}

cout << "Для изначального массива: " << endl;

// Вывод введенного массива

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

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

cout << a[i][k] << " ";

}

cout << endl;

}

//Определение столбца с минимальной суммой

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

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

arr[i] += a[k][i];

}

if (i == 0) {

ma = arr[i];

mc = i;

}

else if (arr[i] < ma) {

ma = arr[i];

mc = i;

}

}

// Сортировка столбца пузырьком

do {

k = 0;

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

if (a[i][mc] < a[i + 1][mc]) {

k = a[i + 1][mc];

a[i + 1][mc] = a[i][mc];

a[i][mc] = k;

k = 1;

}

}

} while (k != 0);

cout << "Изменённый равен: " << endl;

//Вывод изменённого массива

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

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

cout << a[i][k] << " ";

}

cout << endl;

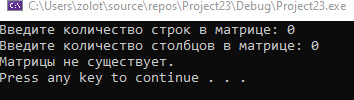
}

system("pause");

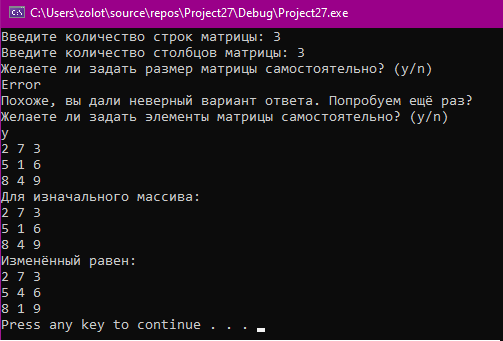
return 0;

}

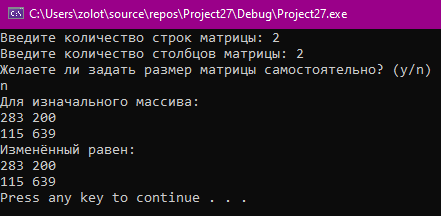
1. Результат выполнения работы программы:

 kjiibibib

ede Отладочный пример №1



Отладочный пример №2



Отладочный пример №3

**6.** Вывод:

В ходе выполнения работы были изучены способы заполнения, обработки двумерных динамических массивов, их выборочную сортировку, а также вывода. Были устранены все исключения, возникшие по ходу работы.

Было проделано исследование способов реализации такой программы с помощью различных учебных материалов

На контрольных примерах мы убедились, что программа работает безошибочно, независимо от введённых данных.

Был оформлен комплект документации на программный код.