МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Физико-технический факультет

Кафедра информатики и вычислительной техники (ИВТ)

Лабораторная работа №06

**Программирование с использованием одномерных массивов**

Выполнил студент 585 гр.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.М. Губченко

Проверил: к.т.н,, доцент каф. ВТиЭ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.Г. Скурыдин

Лабораторная работа защищена

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019 г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Барнаул 2019

1. **Формулировка задачи**

С помощью датчика случайных чисел сформировать двумерный массив n×n и упорядочить (переставить) строки матрицы по не возрастанию сумм элементов строк.

1. **Постановка задачи**

На вход программы поступает целое положительное число n определяющее размерность массива вида n×n. После этого программа находи суммы строк матрицы и сортирует строки по невозрастанию сумм. На выход поступает матрица вида n×n.

1. **Математическая модель**

Для построения математической модели рассмотрим пример. Инициализируется 9-элементный массив вида 3×3, с значениями вида arr(line, column) arr(1, 1)=10, arr(1, 2)=10, arr(1, 3)=10, arr(2, 1)=6, arr(2, 2)= 10, arr(2, 3)=24, arr(3, 1)=2, arr(3, 2)=4, arr(3, 3)=1. После этого считается сумма первой строки равная 30 и сумма второй строки равная 40. Выполняется проверка условия, если сумма первой меньше суммы второй строки, то элементам arr(1, 1), arr(1, 2), arr(1, 3) присваиваются значения arr(2, 1), arr(2, 2) и arr(2, 3), а элементам arr(2, 1) arr(2, 2), arr(2, 3) присваиваются значения arr(1, 1), arr(1, 2), arr(1, 3) соответственно. После этого считается сумма третьей строки равная 7. В данном случае сумма второй строки больше чем сумма третьей строки, значит строки местами не меняются. В конце выполнения алгоритма программа выводит матрицу  
6 10 24

10 10 10

2 4 1

1. **Описание алгоритма**

**Начало алгоритма**

**//**генерация массива

* 1. Ввод переменной n
  2. Присвоение переменной line значения 0
  3. Если line >= n переход к пункту 4.10
  4. Присвоение переменной line значения line++
  5. присвоение переменной column значения 0
  6. Если column >= n переход к пункту 4.3
  7. Присвоение переменной column значения column++
  8. Присвоение переменной arr(line, column) значения (rand() % 20)
  9. Вывод arr(line, column)и переход к пункту 4.6

//сортировка матрицы

* 1. Присвоение переменной sum1 значения 0
  2. Присвоение переменной sum2 значения 0
  3. Присвоение переменной line значения 0
  4. Присвоение переменной q значения 0
  5. Если q >= n переход к пункту 4.40
  6. Присвоение переменной q значения q++

//считаю сумму первой строки

* 1. Присвоение переменной column значения 0
  2. Если column >= n переход к пункту 4.20
  3. Присвоение переменной column значения column++
  4. Присвоение переменной sum1 значения sum1+=arr(line, column) и переход к пункту 4.17
  5. Вывод “sum1= “ sum1

//Считаю сумму второй строки

* 1. Присвоение переменной line значения 1
  2. Если line >= n переход к пункту 4.38
  3. Присвоение переменной line значения line++
  4. Присвоение переменной column значения 0
  5. Если column >= n переход к пункту 4.28
  6. Присвоение переменной column значения column++
  7. Присвоение переменной sum2 значения sum2+= arr(line, column) и переход к пункту 4.25
  8. Вывод “Sum2= “ sum2
  9. Если sum2 < sum1 переход к пункту 4.36
  10. Присвоение переменной a значения line
  11. Присвоение переменной b значения line – 1
  12. Присвоение переменной k значения 0
  13. Если k >= n переход к пункту 4.36
  14. Присвоение переменной k значения k++
  15. Выполнения функции swap(arr(a, k), arr(b, k)) и переход к пункту 4.33
  16. Присвоение переменной sum1 значения sum2
  17. Присвоение переменной sum2 значения 0 и переход к пункту 4.25
  18. Присвоение переменной column значения 0
  19. Присвоение переменной line значения 1 и переход к пункту 4.14

//вывод матрицы

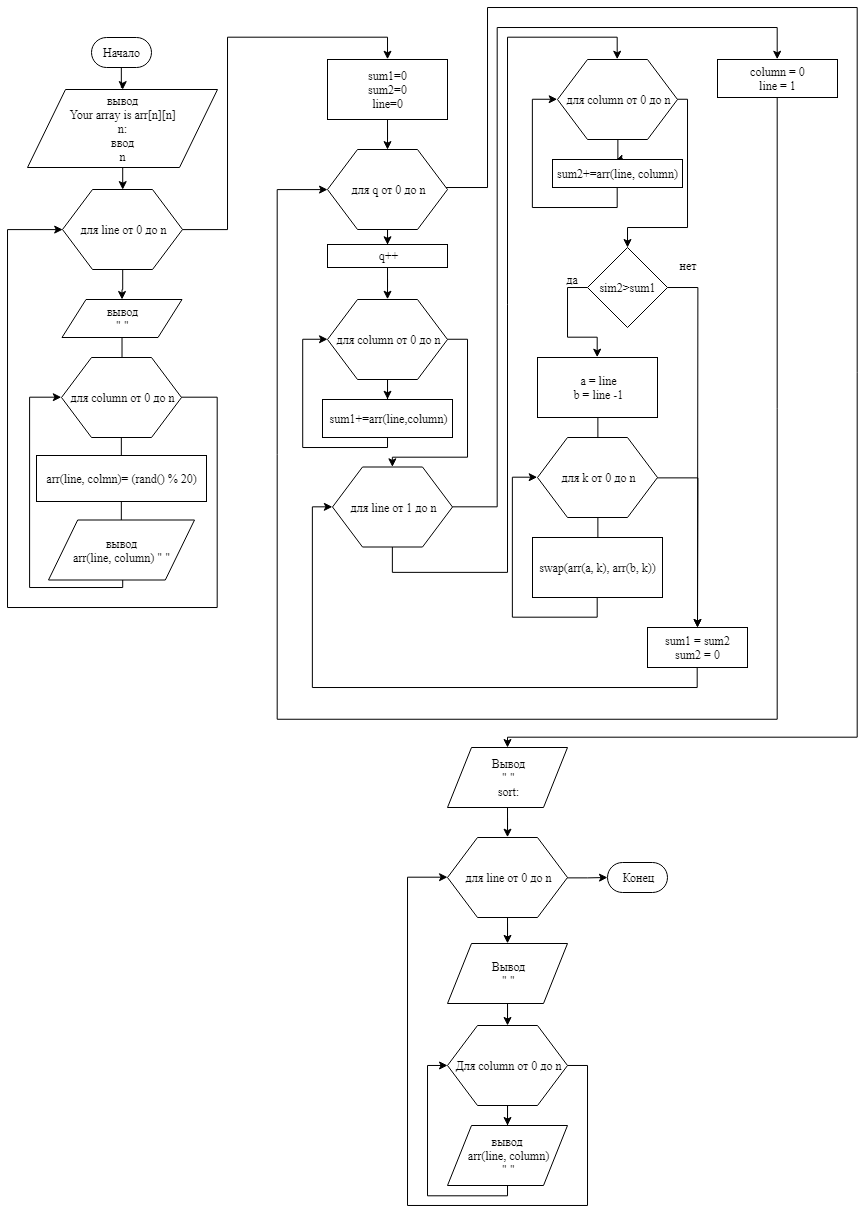
* 1. Вывод “sort:“
  2. Присвоение переменной line значения 0
  3. Если line >=n переход завершение работы программы
  4. Присвоение переменной line значения line++
  5. Вывод “ “
  6. Присвоение переменной column значения 0
  7. Если column >= n переход к пункту 4.42
  8. Присвоение переменной column значения column++
  9. Вывод arr(line, column) “ “ и переход к пункту 4.46

**Конец алгоритма**

1. **Проект программы с сортировкой строк матрицы n**×**n по не убыванию**

В данном случае программа должна состоять из 39 частей: 2 блока вывода, блок ввода, цикл с параметром(блок вывода, блок арифметических действий, цикл с параметром [2 блока арифметических действий, блок вывода]), три блока арифметических действий, цикл с параметром(3 блока арифметических действий, цикл с параметром[2 блока арифметических действий], цикл с параметром[4 блока арифметических действий, цикл с параметром(2 блока арифметических действий) блок проверки условия, цикл с параметром(одна функция)]), 2 блока вывода, цикл с параметром(блок арифметических действий, блок вывода, цикл с параметром[блок арифметических действий, блок вывода])

1. **Опорный граф (блок-схема) алгоритма**



1. **Текст программы:**

**на языке С++**

#include <iostream>

#include <ctime>

#include <cstdlib>

#define arr(line,column) \*(arr+line\*n+column)

using namespace std;

int main()

{

srand(time(NULL));

int n, line, column, sum1, sum2, temp;

//ввод размерности

cout << "Your array is arr[n][n]" << endl;

cout << "n: ";

cin >> n;

//генерация массива

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

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

cout << " " << endl;

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

arr(line, column)= (rand() % 20);

cout << arr(line, column) << " ";

}

}

//сортировка

sum1 = 0;

sum2 = 0;

line = 0;

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

{

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

sum1 += arr(line, column);//считаю сумму чисел первой строки

}

//cout << "sum1= " << sum1 << endl;

for (line = 1; line < n; line++)

{

for (column = 0; column < n; column++)

{

sum2 += arr(line, column);//считаю сумму чисел следующей строки

}

//cout << "Sum2= " << sum2 << endl;

if (sum2 > sum1)

{

int a = line;

int b = line - 1;

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

{

swap(arr(a, k), arr(b, k));

}

}

/\*cout << "-----" << endl;

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

{

cout << " " << endl;

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

cout << arr(line, column) << " ";

}

}

cout << "-----" << endl;\*/

//проверка массивов

sum1 = sum2;

sum2 = 0;

}

column = 0;

line = 1;

}

//вывод массива

cout << " " << endl;

cout << "sort:" << endl;

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

cout << " " << endl;

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

cout << arr(line, column) << " ";

}

}

return 0;

}

1. **Проверка работоспособности (тестирование) программы**

Для проверки работоспособности программы необходимо просто посмотреть на то, что выведет программа в конце работы.

1. **Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду простоты программы она одинаково эффективно выполняется для обоих языков программирования.

1. **Формулировка задачи**

Даны натуральные числа i, j, действительная матрица размера 18×24 (1≤i<j≤24). Поменять в матрица местами i-й и j-й столбцы.

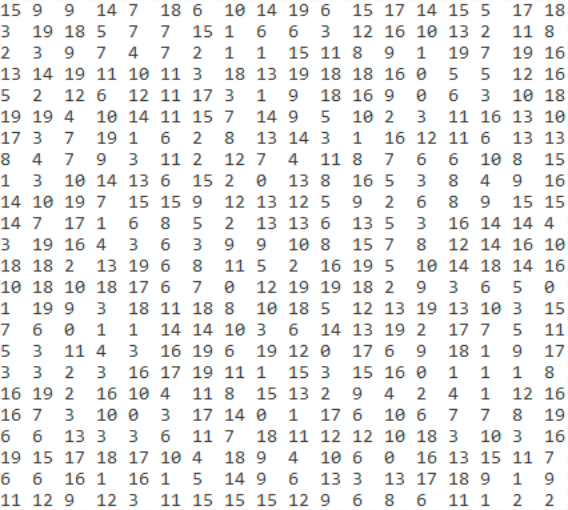
1. **Постановка задачи**

На вход программы поступает целочисленная матрица размерности 18×24 и два вещественных натуральных числа i и j удовлетворяющих условию 1≤i<j≤24. После этого столбец под номером i меняется местами со столбцом под номером j. На выход поступает целочисленная матрица размерности 18×24.

1. **Математическая модель**

Для построения математической модели рассмотрим пример. Инициализируется целочисленная матрица 18×24 

Элементы которой представлены в виде arr(line, column). После этого мы вводим два вещественных, натуральных числа i = 3 и j = 8. Затем элементы вида arr(line, i) меняются местами с элементами вида arr(line, j). После этого мы получим матрицу



1. **Описание алгоритма**

**Начало алгоритма**

* 1. Присвоение переменной \_column значения 18
  2. Присвоение переменной \_line значения 24

//генерация массива

* 1. Присвоение переменной line значения 0
  2. Присвоение переменной line значения line++
  3. Если \_line >= line переход к пункту 4.14
  4. Вывод “ “
  5. Присвоение переменной column значения 0
  6. Если \_column >= column переход к пункту 4.5
  7. Присвоение переменной column значения column++
  8. Присвоение переменной arr(line, column) значения (rand() % 20)
  9. Вывод arr(line, column)
  10. Если arr(line, column) > 9 вывод “ “, иначе вывод “ “
  11. Переход к пункту 4.8

//ввод чисел i и j

* 1. Ввод i
  2. Ввод j

//свап столбцов если i<j

* 1. Если i>=j переход к пункту 4.23
  2. Присвоение переменной line значения 0
  3. Если \_line >= line переход к пункту 4.24
  4. Присвоение переменной line значения line++
  5. Присвоение переменной temp значения arr(line, i)
  6. Присвоение переменной arr(line, i) значения arr(line, j)
  7. Присвоение переменной arr(line, j) значения temp и переход к пункту 4.18
  8. Вывод Error: i>j и конец выполнения программы

//вывод массива

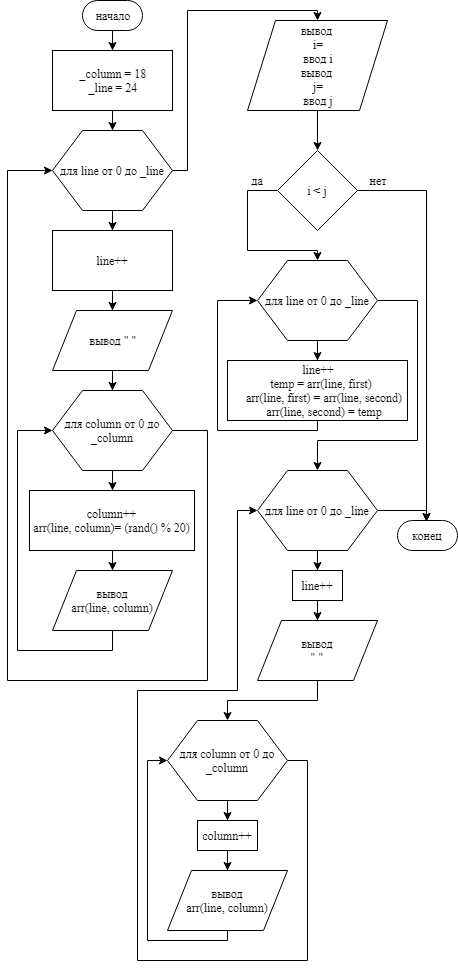
* 1. Присвоение переменной line значения 0
  2. Если \_line>=line конец выполнения программы
  3. Присвоение переменной line значения line++
  4. Вывод “ “
  5. Присвоение переменной column значения 0
  6. Если \_column>=column переход к пункту 4.25
  7. Присвоение переменной column значения column++
  8. Вывод arr(line, column)
  9. Если arr(line, column) > 9 вывод “ “ иначе вывод “ “
  10. Переход к пункту 4.29

**Конец алгоритма**

1. **Проект программы с объединением трёх упорядоченных по убыванию массивов в четвёртый упорядоченный по убыванию массив**

Проект программы с объединением трёх упорядоченных по убыванию массивов в четвёртый упорядоченный по убыванию массив замкнутых. В данном случае программа должна состоять из 44 частей: 2 блока арифметических действий, блок вывода, цикл с параметром(2 блока арифметических действий, блок проверки условия, блок вывода, цикл с параметром[3 блока арифметических действий, 3 блока вывода, 2 блока проверки условия]), 3 блока вывода, 2 блока ввода, блок проверки условия, цикл с параметром(5 блоков арифметических действий, блок проверки условия и блок ввода), цикл с параметром(2 блока арифметических действий, блок вывода, блок проверки условия, цикл с параметром[3 блока арифметических действий, 2 блока проверки условия, 2 блока вывода])

1. **Опорный граф (блок-схема)**



1. **Текст программы:**

**на языке С++**

#include <iostream>

#include <ctime>

#include <cstdlib>

#define arr(line,column) \*(arr+line\*\_line+column)

using namespace std;

int main()

{

srand(time(NULL));

int n, line, column, first, second, temp;

int \_column = 18;

int \_line = 24;

//генерация массива

cout << "Your array" << endl;

int \* arr = (int \*)malloc(\_line \* \_column \* sizeof(int));

for (int line=0; line < \_line; line++)

{

cout << " " << endl;

for (int column =0; column < \_column; column++)

{

arr(line, column)= (rand() % 20);

cout << arr(line, column);

if(arr(line, column) > 9)

{

cout << " ";

} else {

cout << " ";

}

}

}

cout << " " << endl;

cout << "i= ";

cin >> first;

first--;

cout << "j= ";

cin >> second;

second--;

//свап столбцов если условие выполнено

if (first < second)

{

for (int line = 0; line < \_line; line++)

{

temp = arr(line, first);

arr(line, first) = arr(line, second);

arr(line, second) = temp;

}

} else

{

std::cout << "Error: i>j" << '\n';

exit(1);

}

//вывод массива

cout << "sort:" << endl;

for (int line = 0; line < \_line; line++) {

cout << " " << endl;

for (int column = 0; column < \_column; column++) {

cout << arr(line, column);

if(arr(line, column) > 9)

{

cout << " ";

} else {

cout << " ";

}

}

}

return 0;

}

1. **Проверка работоспособности (тестирование) программы**

Для проверки работоспособности программы необходимо посмотреть на то, что выведет программа.

1. **Сравнительный анализ и оценка эффективности работы программ на разных языках программирования**

Ввиду простоты программы она одинаково эффективно выполняется для обоих языков программирования.