МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Физико-технический факультет

Кафедра информатики и вычислительной техники (ИВТ)

Лабораторная работа №07

**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**

Дана строка символов и некоторый символ sym. Сформируйте новую строку, вставив после каждого вхождения символа sym запятую.

1. **Постановка задачи**

На вход программы поступает буквенная строка и символ sym. После этого программа ставит запятую после каждого встреченного символа sym в этой строке. На выход поступает новая строка, полученная после работы программы.

1. **Математическая модель**

Для построения математической модели рассмотрим пример. Инициализируется строка «Hi, how are you doing with the session?» и символ «o». После этого программа проверяет каждый символ строки на совпадение с символом «о». Если встретится символ «о», то программа ставит после него запятую. В конце выполнения алгоритма программа выведет строку «Hi, ho,w are yo,u do,ing with the sessio,n?»

1. **Описание алгоритма**

**Начало алгоритма**

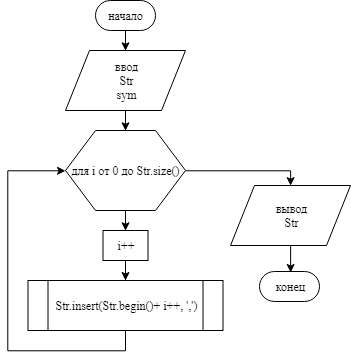
* 1. Ввод строки Str
  2. Ввод символа sym
  3. Присвоение переменной i значения 0
  4. Если i>=Str.size() переход к пункту 4.7
  5. Если (Str[i] != sym), то переход к пункту 4.4
  6. Выполнение Str.insert(Str.begin()+ i++, ',') и переход к пункту 4.4
  7. Вывод строки Str

**Конец алгоритма**

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

В данном случае программа должна состоять из 9 частей: 2 блока ввода, цикл с параметром[три блока арифметических действий, два блока проверки условий, стандартная функция insert()]

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



1. **Текст программы:**

**на языке С++**

#include <iostream>

#include <string>

**using** **namespace** std;

**int** main()

{

**char** sym;

string Str;

cout << "enter the string:" << '\n';

getline(cin, Str);

cout << "enter character: ";

cin >> sym;

**for**(size\_t i=0; i<Str.size(); i++)

{

**if**(Str[i] == sym)

{

Str.insert(Str.begin()+ i++, ',');

}

}

cout<<Str<<'\n';

**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.

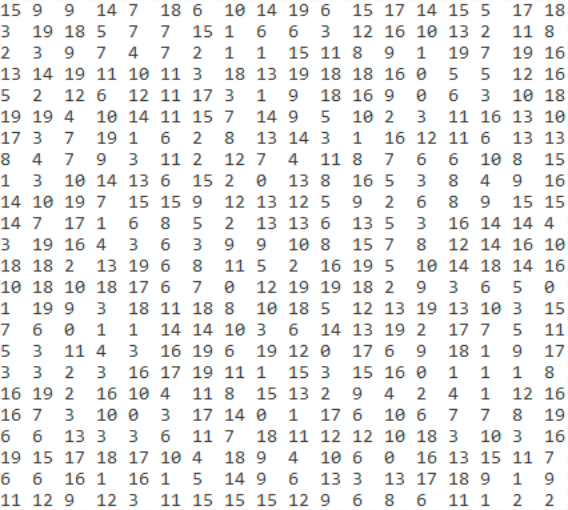
1. **Постановка задачи**

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

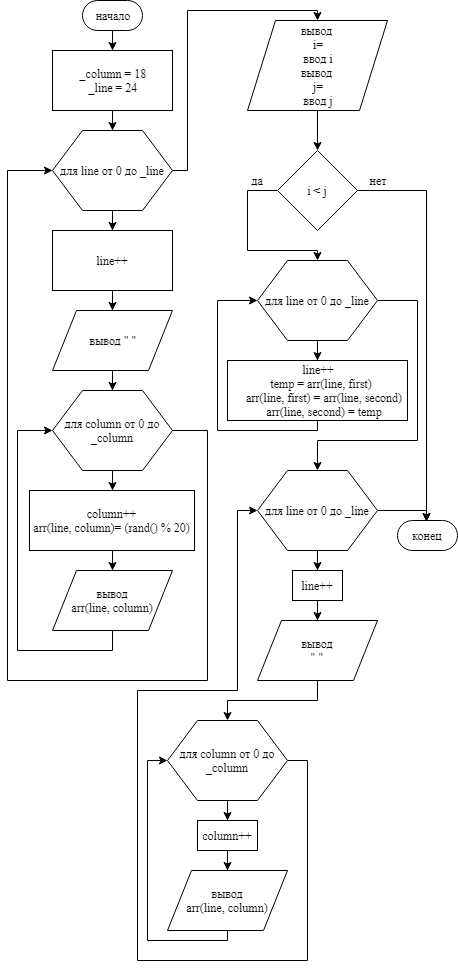
* 1. Присвоение переменной line значения 0
  2. Если \_line>=line конец выполнения программы
  3. Присвоение переменной line значения line++
  4. Вывод “ “
  5. Присвоение переменной column значения 0
  6. Если \_column>=column переход к пункту 4.25
  7. Присвоение переменной column значения column++
  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.