МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Лабораторная работа №2\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование темы проекта или работы)

ОТЧЁТ

По лабораторной работе

по дисциплине

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Алгоритмы и структуры данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование дисциплины)

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2022

**Задание:**

Дана разреженная матрицы (CCS). Переставить столбцы в матрице по

возрастанию сумм элементов в этих столбцах.

CCS - Разреженный столбцовый формат

**Программный код:**

#include <iostream>

#include <fstream>

#include <cmath>

#include <iomanip>

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

int main(){

char logo[] =

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

"\* Nizhny Novgorog State Technical University \*\n"

"\* Study work number 2. Task number 1. \*\n"

"\* Performed student 21-IVT-2 Kitov A. A. \*\n"

"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

std::cout<< logo <<std::endl;

int k=0,n=0;

int num=0;

std::ifstream in1;

in1.open("output.txt");

//блок подсчёта количества столбцов в матрице и количества ненулевых элементов

while(!in1.eof()){

in1>>num;

if(num!=0){

n=n+1;

}

k=k+1;

}

in1.close();

k=k-1;

k=sqrt(k);

std::ifstream in2;

in2.open("output.txt");

int\* N;

int freenum=0, freenum2=0;

//блок создания динамических массивов

N=new int [n]; /////////////////////////////////////////////////////

for(int g=0;g<n;g++){ // Динамический массив номеров ненулевых элементов //

N[g]=g; /////////////////////////////////////////////////////

}

int\* SUM; //////////////////////////////////////////

SUM=new int [k]; // Динамический массив для сумм стобцов //

for(int g=0;g<k;g++){ //

SUM[g]=0; // Заполнение массива нулями //

} //////////////////////////////////////////////////////

int\* A; // Динамический массив хранения ненулевых элементов //

A=new int [n]; //////////////////////////////////////////////////////

int\* LI; // Динамический массив хранения строки элемента //

LI=new int [n]; //////////////////////////////////////////////////////////

int\* LJ; // Динамический массив хранения номера первого элемента //

LJ=new int [k+1]; //////////////////////////////////////////////////////////

int\*\* TXT; // //

TXT=new int\* [k]; // Двумерный динамический массив для работы с матрицей //

for(int g=0;g<k;g++){ // элементов //

TXT[g]=new int [k]; // //

} /////////////////////////////////////////////////////////

// заполнение матрицы из файла

for(int g=0;g<k;g++){

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

in2>>TXT[g][i];

}

}

//вывод матрицы из файла

for(int g=0;g<k;g++){

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

std::cout<<std::setw(4)<<TXT[g][i]<<" ";

}

std::cout<<std::endl;

}

bool flag=true;

int fr=1;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

// Блок заполнения в формате css //

for(int g=0;g<k;g++){

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

if(TXT[i][g]!=0){

A[freenum]=TXT[i][g]; //заполение массива элементов

LI[freenum]=i; //заполнение массива номерами строк

if(flag==true){

LJ[freenum2]=fr; //заполнение массива номерами первых в столбцах элементов

flag=false;

}

freenum=freenum+1;

fr=fr+1;

SUM[g]=SUM[g]+TXT[i][g];

}

}

freenum2=freenum2+1;

flag=true;

}

for(int g=0;g<k;g++){

N[g]=g;

}

std::cout<<std::endl;

LJ[k]=n;

in2.close();

for(int g=0;g<k;g++){

if(LJ[g]==0 and g+1<k){

LJ[g]=LJ[g+1];

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

// Блок вывода итоговых массивов //

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

if(i==0){

std::cout<<"N: ";

}

std::cout<<std::setw(4)<<N[i]+1<<" ";

}

std::cout<<std::endl;

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

if(i==0){

std::cout<<"A: ";

}

std::cout<<std::setw(4)<<A[i]<<" ";

}

std::cout<<std::endl;

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

if(i==0){

std::cout<<"LI: ";

}

std::cout<<std::setw(4)<<LI[i]+1<<" ";

}

std::cout<<std::endl;

for(int i=0;i<k+1;i++){

if(i==0){

std::cout<<"LJ: ";

}

std::cout<<std::setw(4)<<LJ[i]<<" ";

}

std::cout<<std::endl;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

int maxx=0, lastg=0;

int\* MaxKoef;

MaxKoef=new int [k];

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

// Блок вывода сумм столбцов матрицы по убыв//

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

for(int g=0;g<k;g++){

if(SUM[g]>maxx){

maxx=SUM[g];

MaxKoef[i]=g+1;

}

lastg=g;

}

for(int g=0;g<k;g++){

if(maxx==SUM[g]){

SUM[g]=-2;

break;

}

}

if(i==0){

std::cout<<std::endl<<"Суммы: ";

}

if(maxx>=0){

std::cout<<std::setw(4)<<maxx<<" ";

}

maxx=-1;

}

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

if(i==0){

std::cout<<std::endl<<"Столбец: ";

}

std::cout<<std::setw(4)<<MaxKoef[i]<<" ";

}

std::cout<<std::endl;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

return 0;

}

**Результаты работы программы:**





