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

Списки. Стеки. Очереди.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к отчету по лабораторной работе №2

по дисциплине

Алгоритмы и структуры данных

РУКОВОДИТЕЛЬ:

Капранов Сергей Николаевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

СТУДЕНТ:

Семехин Василий Дмитриевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

19-ИВТ-3

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

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

Нижний Новгород 2020

**Задача 19**

Дана разреженная матрица CRS. Найти количество её различных элементов и вывести их на экран.

**Реализация кода на языке С++**

**Функция генерации матрицы**

**первого типа**

//Функиця генерации матрицы 4х4

void generate1(Matrix& matrix, vctr& A, vctr& LI, vctr& LJ, int size) {

for (int i = 0; i < size; i++) { //Создаем матрицу нулевых элементов

vector<int> v(4, 0);

for (int j = 0; j < size; j++) { //Заполняем будующую матрицу нулями

matrix.push\_back(v);

}

}

for (int i = 0; i < size; i++) { //Задаем ненулевые элементы матрицы

//Псевдо-случайным образом

int rndm = rand() % 3; //Переменна, генерирующая количество

//Ненулевых элементов в строке

//(от 0 до 2)

int pos = rand() % 3; //Генератор случайной позиции элемента

if (rndm == 1) { //В зависимости от количества

/Ненулевых значений

//Элементы с указанными индексами

//Объявляются

matrix[i][pos] = rand() % 8 + 1;

}

if (rndm == 2) {

matrix[i][pos] = rand() % 8 + 1;

matrix[i][3] = rand() % 8 + 1;

}

}

PrintMatrix(matrix, size); //Вызываем функцию отображения матрицы

int NNZ = 0; //Количество ненулевых элементов

int zero = 0; //Количество нулевых элементов

while (NNZ == 0) { //Данный цикл необходим для того,

//чтобы избежать появления

//полностью нулевой матрицы

for (int i = 0; i < size; i++) { //Двойной цикл для поиска ненулевых

//Элементов

for (int j = 0; j < size; j++) {

if (matrix[i][j] != 0) {

A.push\_back(matrix[i][j]); //Добавляем ненулевые элементы матрицы

//В массив А

LJ.push\_back(j + 1); //Добавляем номер столбца, в котором //Находится элемент

//В массив LJ

NNZ++; //Считаем количество ненулевых

//Элементов в i-ой строке

}

else {

zero++; //Считаем количество нулевых элементов

}

}

LI.push\_back(NNZ); //После занесения всех ненулевых

//Элементов

//В массив А, в массив LI заносится

//диапазон их позиций в массиве А

}

}

PrintVector(A, LI, LJ); //Отображаем получившуюся CRS

//Отображаем количество различных

//Элементов

cout << "Value of non zero elements: " << NNZ << endl;

cout << "Value of zero elements: " << zero << endl;

}

**Функция отображения матрицы**

//Функция, отображающая матрицу

void PrintMatrix(Matrix matrix, int size){ //Передаем в функцию матрицу М

for (int i = 0; i < size; i++) { //Отображение матрицы

cout << " | ";

for (int j = 0; j < size; j++){

cout << matrix[i][j] << " ";

}

cout << "|" << endl;

}

cout << endl;

}

**Функция отображения векторов**

//Функция, отображающая CRS

void PrintVector(vctr A, vctr LI, vctr LJ){

cout << "N = [ ";

for (int i = 1; i <= A.size(); i++) { //Отображение количества ненулевых

//Элементов

//В виде таблицы

cout << i << " ";

}

cout << "]" << endl;

cout << "A = [ "; //Отображение массива А

for (int i = 0; i < A.size(); i++){

cout << A[i] << " ";

}

cout << " ]" << endl;

cout << "LJ = [ "; //Отображение массива LJ

for (int i = 0; i < LJ.size(); i++) {

cout << LJ[i] << " ";

}

cout << " ]" << endl;

cout << "LI = [ "; //Отображение массива LI

for (int i = 0; i < LI.size(); i++) {

cout << LI[i] << " ";

}

cout << " ]" << endl;

}

**Функция генерации матрицы 2 типа**

//Функиця генерации матрицы 5х5

void generate2(Matrix &matrix,vctr &A, vctr &LI, vctr &LJ, int size) {

for (int i = 0; i < size; i++) { //В функции generate2

//Использованы те же

//Алгоритмы, что и в функции

//generate1, не считая

//Разницы в размерах матриц

vector<int> v(5, 0);

for (int j = 0; j < size; j++) {

matrix.push\_back(v);

}

}

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

int rndm = rand() % 3;

int pos = rand() % 3;

if (rndm == 1) {

matrix[i][pos] = rand() % 8 + 1;

}

if (rndm == 2) {

matrix[i][pos] = rand() % 8 + 1;

matrix[i][4] = rand() % 8 + 1;

}

}

PrintMatrix(matrix, size);

int NNZ = 0;

int zero = 0;

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

for (int j = 0; j < size; j++) {

if (matrix[i][j] != 0) {

A.push\_back(matrix[i][j]);

LJ.push\_back(j+1);

NNZ++;

}

else {

zero++;

}

}

LI.push\_back(NNZ);

}

PrintVector(A, LI, LJ);

cout << "Value of non zero elements: " << NNZ << endl;

cout << "Value of zero elements: " << zero << endl;

}

**Функция генерации матрицы 3 типа**

//Функиця генерации матрицы 6х6

void generate3(Matrix& matrix, vctr& A, vctr& LI, vctr& LJ, int size) {

for (int i = 0; i < size; i++) { //В функции generate2 использованы

//Практически

//Те же алгоритмы, что и в функции

//generate1. Разница заключается

//В размерах матриц

vector<int> v;

for (int j = 0; j < size; j++) {

v.push\_back(0);

}

matrix.push\_back(v);

}

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

int rndm = rand() % 3; //Количестве ненулевых элементов

/В строке

int pos = rand() % 5; //И их позиции в матрице

if (rndm == 1) {

matrix[i][pos] = rand() % 8 + 1;

}

if (rndm == 2) {

matrix[i][pos] = rand() % 8 + 1;

matrix[i][5] = rand() % 8 + 1;

}

if (rndm == 3) {

matrix[i][0] = rand() % 8 + 1;

matrix[i][2] = rand() % 8 + 1;

matrix[i][4] = rand() % 8 + 1;

}

}

PrintMatrix(matrix, size);

int NNZ = 0;

int zero = 0;

while (NNZ == 0) {

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

for (int j = 0; j < size; j++) {

if (matrix[i][j] != 0) {

A.push\_back(matrix[i][j]);

LJ.push\_back(j + 2);

NNZ++;

}

else {

zero++;

}

}

LI.push\_back(NNZ);

}

}

PrintVector(A, LI, LJ);

cout << "Value of non zero elements: " << NNZ << endl;

cout << "Value of zero elements: " << zero << endl;

}

**Функция main**

int main() {

srand(time(0));

int s; //Переменная для определения размеров

/Мматрицы пользователем

const int size1 = 4, size2 = 5, size3 = 6; //Возможные размеры матрицы

vctr A,LI,LJ; //Объявление CRS

Matrix matrix; //Обявление разряженной матрицы

cout << "Hello! This`s CRS. Choose size of matrix (3x3, 4x4, 5x5)." << endl;

cout << "Enter 1 for (4x4), 2 for (5x5) and 3 for (6x6) matrix: ";

//Проверка вводимых значений

while (!(cin >> s) || (cin.peek() != '\n') || s < 1 || s > 3)

{

cin.clear();

while (cin.get() != '\n');

cout << "Error, enter correct value!" << endl;

}

cout << endl;

if (s == 1){ //В зависимости от выбранного

//Пользователем размера

//Вызываются различные функции для

//Генерации CRS

generate1(matrix, A, LI, LJ, size1);

}

if (s == 2) {

generate2(matrix, A, LI, LJ, size2);

}

if (s == 3) {

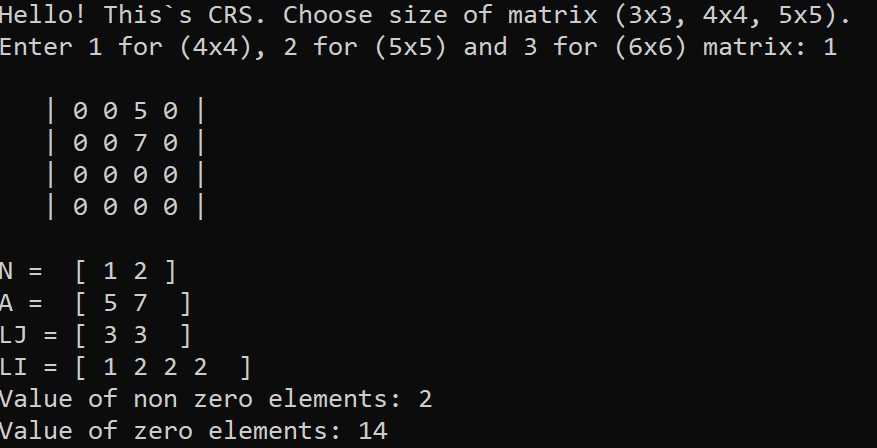
generate3(matrix, A, LI, LJ, size3);

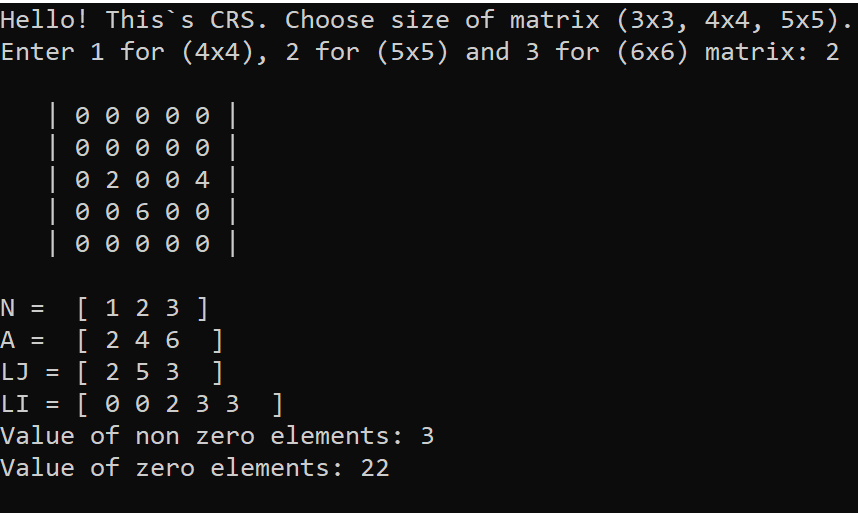
}

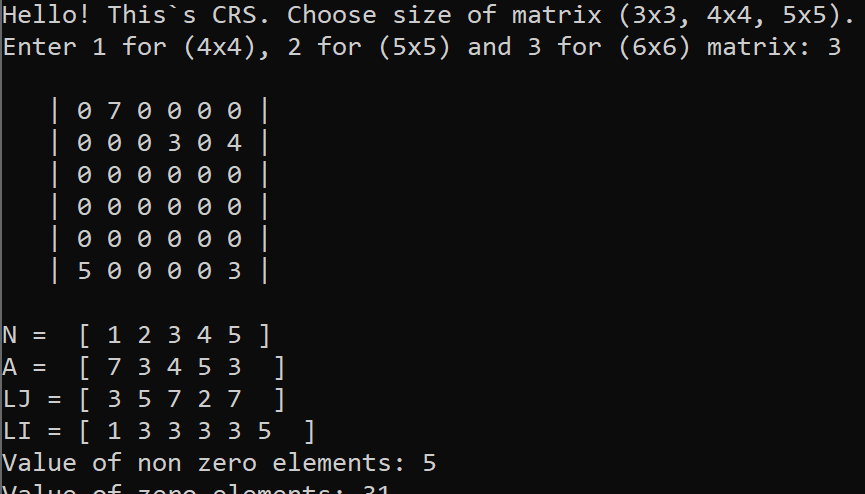
return 0;

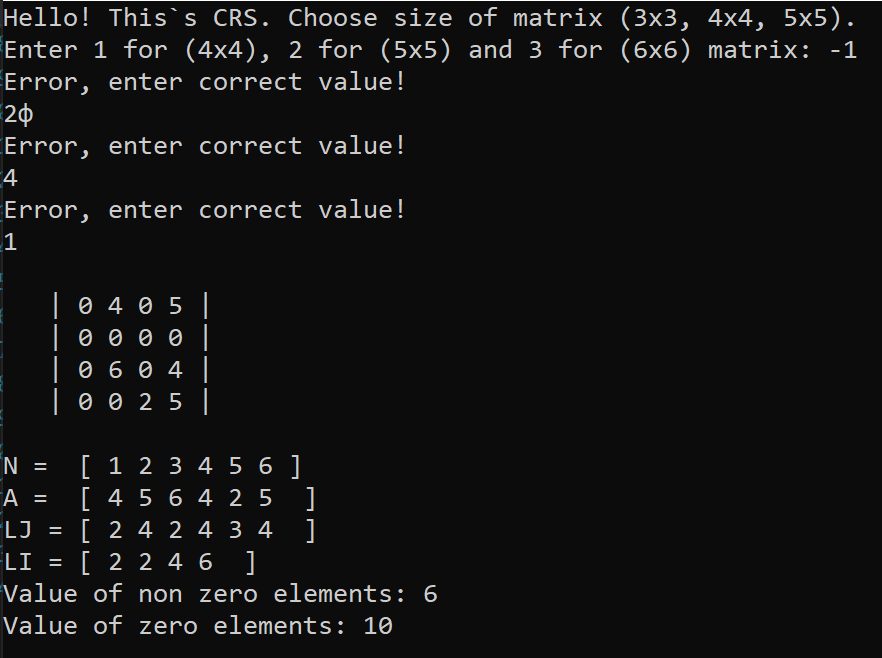
}

**Результат работы программы**



****

****

****