**Отчет по лабораторной работе № 7** по курсу

«Языки и методы программирования»

Студент группы М8О-112Б-21 Орешкин Максим Алексеевич, № по списку \_\_\_13\_\_

Контакты www, e-mail: maks-oreh03@mail.ru

Работа выполнена: « » \_\_\_\_\_\_ 202\_ г.

Преподаватель: доцент каф. 806 \_Никулин С.П\_\_\_\_\_\_\_\_\_\_\_\_\_

Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан « » \_\_\_\_\_\_\_\_\_202\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ь преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Тема:** Разреженные матрицы
2. **Цель работы:** Составить программу на языке Си с процедурами и/или функциями для обработки прямоугольных разреженных матриц с элементами целого типа, которая:

1. вводит матрицы различного размера, представленные во входном текстовом файле в обычном формате (по строкам), с одновременным размещением ненулевых элементов в разреженной матрице в соответствии с заданной схемой;

2. печатает введенные матрицы во внутреннем представлении согласно заданной схеме размещения и в обычном (естественном) виде;

3. выполняет необходимые преобразования разреженных матриц (или вычисления над ними) путем обращения к соответствующим процедурам и/или функциям;

4. печатает результат преобразования (вычисления) согласно заданной схеме размещения и в обычном виде

1. **Задание** (*вариант № 1* )**:**

**Варианты схемы размещения матрицы:**

1. Цепочка ненулевых элементов в векторе A со строчным индексированием (индексы в массиве M равны 0, если соответствующая строка матрицы содержит только нули)



Индекс, равный нулю, означает отсутствие ненулевых элементов в строке (или в ее остатке). Если матрицы не изменяются программой, возможна экономия памяти за счет отказа от хранения в массиве А индексов следующего элемента столбца (когда элементы идут подряд). Вставка и удаление при этом способе возможны, но чересчур дороги: число перестановок элементов составит O(N) вместо O(1).

Реализация на динамических структурах — векторах. Вариант преобразований — нахождение максимального по модулю элемента и деление на него элементов строки, в которой он находится.

1. **Оборудование** :

ЭВМ , процессор , имя узла сети с ОП ГБ,

НМД ГБ. Терминал адрес . Принтер

Другие устройства

*Оборудование ПЭВМ студента, если использовалось:*

 Процессор intel core i7, с   ОП 8 ГБ, НМД  120832 МБ. Монитор  15,6 /1920\*1820

Другие устройства

1. **Программное обеспечение:**

Операционная система семейства , наименование версия интерпретатор команд версия

Система программирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Редактор текстов версия

Утилиты операционной системы

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

Прикладные системы и программы

Местонахождение и имена файлов программ и данных

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Unix , наименование Ubuntu версия 20.04

интерпретатор команд bash версия 4.4.18

Система программирования версия

Редактор текстов vim версия 8.1

Утилиты операционной системы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладные системы и программы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение и имена файлов программ и данных на домашнем компьютере \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Идея, метод, алгоритм**  решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Программа имеет меню, в котором пользователь может выбрать действие, которое будет делать программа:

1. Вводит матрицуразмера, который задаётся в начале выполнения программы пользователем, из входного текстового файла в соответствии с заданной схемой.
2. Печатает матрицу во внутреннем представлении.
3. Печатает матрицу в естественном представлении.
4. Обрабатывает матрицу согласно заданию.
5. Заканчивает работу.

Программа использует библиотеки «int\_vector.h» и «double\_vector.h», написанные мной же для работы с векторами различных типов.

1. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

Основная программа codec.c:

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <math.h>

#include "int\_vector.h"

#include "double\_vector.h"

int main () {

int str, stlb, term = 0, action, count, first = 1;

double max;

int\_vector m, ind;

create\_int\_vector (&m, str);

create\_int\_vector (&ind, str);

double\_vector a;

create\_double\_vector (&a, 0);

printf ("Вас приветствует программа по обработке разреженных матриц. Введите размер матриц, с которыми будет работать программа в формате: количество строк, количество стобцов.\n");

scanf ("%d %d", &str, &stlb);

while (term != 1) {

printf ("Выберите действие:\n1) Ввести матрицу.\n2) Распечатать матрицу во внутреннем представлении.\n3) Распечатать матрицу в естественном виде.\n4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.\n5) Выход.\n");

scanf ("%d", &action);

switch (action) {

case 1:

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

push\_int\_vector (&m, i, 0);

}

count = 0;

if (first == 0) {

kill\_double\_vector (&a);

create\_double\_vector (&a, 0);

}

int new\_str = -1;

printf ("Для ввода матрицы введите название файла, в котором записаны элементы матрицы подряд через разделители по строкам. Учтите, что файл должен находитсья в той же директории, что и исполяемый файл a.out\n");

char name [20];

scanf ("%s", name);

FILE\* file = fopen (name, "r");

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

new\_str = -1;

if ((count != 0) && (read\_int\_vector (&m, i - 1) != 0)) push1\_double\_vector (&a, 0.0);

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

double el;

fscanf (file, "%lf", &el);

if (el != 0) {

if (new\_str == -1) {

push\_int\_vector (&m, i, (count \* 3 + 1));

push2\_double\_vector (&a, (double)(j + 1), el);

new\_str = 0;

count++;

}

else {

push3\_double\_vector (&a, (double)(count \* 3 + 1), (double)(j + 1), el);

count++;

}

}

}

}

push1\_double\_vector (&a, 0.0);

fclose (file);

first = 0;

break;

case 2:

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

printf ("%d ", read\_int\_vector (&m, j));

}

printf ("\n");

for (int i = 0; i < count \* 3 ; i++) {

double el = read\_double\_vector (&a, i);

if (el - (int)el == 0) {

printf ("%d ", (int)el);

}

else {

printf ("%.1lf ", el);

}

}

printf ("\n");

break;

case 3:

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

int s = read\_int\_vector (&m, i);

if (s == 0) {

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

printf ("0 ");

}

printf ("\n");

}

else {

int schet = 1;

while (read\_double\_vector (&a, s + 1) != 0) {

if (schet != read\_double\_vector (&a, s - 1)) {

printf ("0 ");

schet++;

}

else {

printf ("%.1lf ", read\_double\_vector (&a, s));

schet++;

s = (int)read\_double\_vector (&a, s + 1);

}

}

if (schet == 1) {

for (int l = 0; l < stlb; l++) {

if (schet != read\_double\_vector (&a, s - 1)) {

printf ("0 ");

schet++;

}

else {

printf ("%.1lf ", read\_double\_vector (&a, s));

schet++;

}

}

}

else {

if (read\_double\_vector (&a, s + 1) == 0) {

printf ("%.1lf ", read\_double\_vector (&a, s));

schet++;

}

}

for (int j = schet; j <= stlb; j++) {

printf ("0 ");

}

printf ("\n");

}

}

break;

case 4:

max = 0;

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

push\_int\_vector (&ind, i, 0);

}

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

int s = read\_int\_vector (&m, j), gift = 1;

if (s != 0) {

while (gift) {

if (read\_double\_vector (&a, s + 1) != 0) {

if (abs (read\_double\_vector (&a, s)) > abs(max)) {

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

push\_int\_vector (&ind, i, 0);

}

push\_int\_vector (&ind, j, 1);

max = read\_double\_vector (&a, s);

}

if (abs(read\_double\_vector (&a, s)) == abs(max)) {

push\_int\_vector (&ind, j, 1);

}

s = (int)read\_double\_vector (&a, s + 1);

}

else {

if (abs (read\_double\_vector (&a, s)) > abs(max)) {

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

push\_int\_vector (&ind, i, 0);

}

push\_int\_vector (&ind, j, 1);

max = read\_double\_vector (&a, s);

}

if (abs(read\_double\_vector (&a, s)) == abs(max)) {

push\_int\_vector (&ind, j, 1);

}

s = (int)read\_double\_vector (&a, s + 1);

gift = 0;

}

}

}

}

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

int i = read\_int\_vector (&ind, j), s = read\_int\_vector (&m, j);

if (i) {

int gift = 1;

while (gift != 0) {

if (read\_double\_vector (&a, s + 1) == 0) {

push\_double\_vector (&a, s, read\_double\_vector (&a, s) / max);

gift = 0;

}

else {

push\_double\_vector (&a, s, read\_double\_vector (&a, s) / max);

s = (int)read\_double\_vector (&a, s + 1);

}

}

}

}

break;

case 5:

term = 1;

kill\_double\_vector (&a);

kill\_int\_vector (&m);

kill\_int\_vector (&ind);

}

}

}

Библиотеки с векторами:

«int\_vector.h»:

#ifndef \_\_int\_vector\_h\_\_

#define \_\_int\_vector\_h\_\_

typedef struct {

int len;

int\* data;

} int\_vector;

void create\_int\_vector (int\_vector\* v, int len) {

v->len = len;

v->data = malloc (sizeof (int) \* v->len);

};

void push\_int\_vector (int\_vector\* v, int i, int el) {

v->data [i] = el;

};

int read\_int\_vector (int\_vector\* v, int i) {

return v->data [i];

};

void kill\_int\_vector (int\_vector\* v) {

v->len = 0;

free (v->data);

};

#endif

«double\_vector.h»:

#ifndef \_\_double\_vector\_h\_\_

#define \_\_double\_vector\_h\_\_

typedef struct {

int len;

double\* data;

} double\_vector;

void create\_double\_vector (double\_vector\* v, int len) {

v->len = len;

v->data = malloc (sizeof (double) \* v->len);

};

void push\_double\_vector (double\_vector\* v, int i, double el) {

v->data [i] = el;

};

void push1\_double\_vector (double\_vector\* v, double el) {

v->len += 1;

v->data = realloc (v->data, sizeof (double) \* v->len);

v->data [v->len - 1] = el;

};

void push2\_double\_vector (double\_vector\* v, double el1, double el2) {

v->len += 2;

v->data = realloc (v->data, sizeof (double) \* v->len);

v->data [v->len - 2] = el1;

v->data [v->len - 1] = el2;

};

void push3\_double\_vector (double\_vector\* v, double el1, double el2, double el3) {

v->len += 3;

v->data = realloc (v->data, sizeof (double) \* v->len);

v->data [v->len - 3] = el1;

v->data [v->len - 2] = el2;

v->data [v->len - 1] = el3;

};

void push4\_double\_vector (double\_vector\* v, double el1, double el2, double el3, double el4) {

v->len += 4;

v->data = realloc (v->data, sizeof (double) \* v->len);

v->data [v->len - 4] = el1;

v->data [v->len - 3] = el2;

v->data [v->len - 2] = el3;

v->data [v->len - 1] = el4;

};

double read\_double\_vector (double\_vector\* v, int i) {

return v->data [i];

};

int len\_double\_vector (double\_vector\* v) {

return v->len;

};

void kill\_double\_vector (double\_vector\* v) {

v->len = 0;

free (v->data);

};

#endif

*Пункты 1-7 отчета составляются строго до начала лабораторной работы.*

*Допущен к выполнению работы.*  **Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **Распечатка протокола**  (подклеить листинг окончательного варианта программы с тестовыми примерами, подписанный преподавателем).

maxim@vb:~$cat < matr1   
1.1 0 0 0 0   
0 0 0 0 0    
0 0 2.2 3.3 4.4   
0 0 0 5.5 0    
0 0 0 6.6 0   
maxim@vb:~$cat < matr2   
0 0 0 0 0    
0 0 0 -200 0   
0 0 0   0 1   
0 0 44 5 23.5   
0.1 9 0 0 0

Вас приветствует программа по обработке разреженных матриц. Введите размер матриц, с которыми будет работать программа в формате: количество строк, количество стобцов.   
5 5   
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
1   
Для ввода матрицы введите название файла, в котором записаны элементы матрицы подряд через разделители по строкам. Учтите, что файл должен находитсья в той же директории, что и исполяемый файл a.out   
matr1   
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
2   
1 0 4 13 16    
1 1.1 0 3 2.2 7 4 3.3 10 5 4.4 0 4 5.5 0 4 6.6 0    
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
3   
1.1 0 0 0 0    
0 0 0 0 0    
0 0 2.2 3.3 4.4    
0 0 0 5.5 0    
0 0 0 6.6 0    
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
4   
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
3   
1.1 0 0 0 0    
0 0 0 0 0    
0 0 2.2 3.3 4.4    
0 0 0 5.5 0    
0 0 0 1.0 0    
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
1   
Для ввода матрицы введите название файла, в котором записаны элементы матрицы подряд через разделители по строкам. Учтите, что файл должен находитсья в той же директории, что и исполяемый файл a.out   
matr2   
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
2   
0 1 4 7 16    
4 -200 0 5 1 0 3 44 10 4 5 13 5 23.5 0 1 0.1 19 3 9 0    
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
3   
0 0 0 0 0    
0 0 0 -200.0 0    
0 0 0 0 1.0    
0 0 44.0 5.0 23.5    
0.1 9.0 0 0 0    
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
4   
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
3   
0 0 0 0 0    
0 0 0 1.0 0    
0 0 0 0 1.0    
0 0 44.0 5.0 23.5    
0.1 9.0 0 0 0    
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
5

maxim@vb:~$cat < matr3   
0 0 0 0 0 0 0 0 0 0    
0 0 0 0 0 50.5 0 0 1 2   
0 1.6 7.5 0 0 0 0 0 0 0    
0 0 0 0 50.5 30.9 0 0 0 0  
1.6 0 0 0 0 0 0 0 0 0    
maxim@vb:~$./a.out   
Вас приветствует программа по обработке разреженных матриц. Введите размер матриц, с которыми будет работать программа в формате: количество строк, количество стобцов.   
5 10   
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
1   
Для ввода матрицы введите название файла, в котором записаны элементы матрицы подряд через разделители по строкам. Учтите, что файл должен находитсья в той же директории, что и исполяемый файл a.out   
matr3   
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
2   
0 1 10 16 22    
6 50.5 4 9 1 7 10 2 0 2 1.6 13 7 7.5 0 5 50.5 19 10 30.9 0 1 1.6 0    
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
3   
0 0 0 0 0 0 0 0 0 0    
0 0 0 0 0 50.5 0 0 1.0 2.0    
0 1.6 7.5 0 0 0 0 0 0 0    
0 0 0 0 50.5 30.9 0 0 0 0    
1.6 0 0 0 0 0 0 0 0 0    
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
4   
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
3   
0 0 0 0 0 0 0 0 0 0    
0 0 0 0 0 1.0 0 0 0.0 0.0    
0 1.6 7.5 0 0 0 0 0 0 0    
0 0 0 0 1.0 0.6 0 0 0 0    
1.6 0 0 0 0 0 0 0 0 0    
Выберите действие:   
1) Ввести матрицу.   
2) Распечатать матрицу во внутреннем представлении.   
3) Распечатать матрицу в естественном виде.   
4) Выполнить действие: 'поиск максимального по модулю элемента и деление на него всех элементов строки, в которой он находится'.   
5) Выход.   
5

1. **Дневник отладки** должен содержать дату и время сеансов отладки и основные события (ошибки в сценарии и программе, нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб. или дом. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. **Замечания автора** по существу работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 11. **Выводы**

За выполнение КР я научился работать с разреженными матрицами\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

Подпись студента\_\_\_\_\_\_\_\_\_\_\_\_