**Отчет по лабораторной работе № VII** по курсу Практикум на ЭВМ

Студент группы М8О-110Б-21 Агеева Алиса № по списку 2

Контакты e-mail, telegram, skype alisa.ageewa2003@gmail.com

Работа выполнена: « » \_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

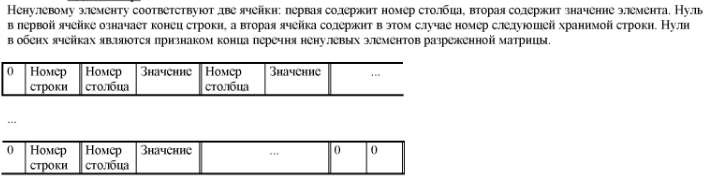
Преподаватель: доцент каф. 806 Никулин Сергей Петрович

Отчет сдан « » \_\_\_\_\_\_\_\_\_20 \_\_\_ г., итоговая оценка \_\_\_\_\_

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

* 1. **Тема:** Разреженные матрицы.

1. **Цель работы:** Составить программу на языке Си с процедурами и/или функциями для обработки прямоугольных разреженных матриц с элементами целого типа.
2. **Задание** (*вариант №* **2** ) **: О**пределить максимальный элемент матрицы и разделить на него все элементы столбца, в котором он находится. Если их несколько – то проделать задание для последнего.
3. **Схема размещения:** Один вектор.



**Физическое представление:** Отображение на массив.

1. **Оборудование** (лабораторное):

ЭВМ \_\_\_\_\_\_\_\_\_\_\_\_\_\_, процессор \_\_\_\_\_\_\_\_\_\_\_\_\_, имя узла сети \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ с ОП \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мб, НМД \_\_\_\_\_\_\_\_\_\_ Мб. Терминал \_\_\_\_\_\_\_\_\_\_ адрес \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. Принтер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Другие устройства

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel Core i3 с ОП 4 Гб НМД 128 Гб. Монитор 1920x1080~60Hz

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Программное обеспечение (лабораторное):**

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 Linux , наименование Ubuntu версия 20.04.1

интерпретатор команд GNU bash версия 5.0.17 .

Система программирования Code::Blocks IDE версия 20.03

Редактор текстов emacs версия 25.2.2

Утилиты операционной системы gcc, cat

Прикладные системы и программы

Местонахождение и имена файлов программ и данных на домашнем компьютере \

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

1. Заведем структуру матрицы Matrix,в которой будем хранить размеры матрицы, вектор и количество ненулевых элементов
2. Опишем следующие процедуры:

void menu() - печатает меню возможных команд.

void print(Matrix a) — печатает матрицу в нормальном виде.

1. Опишем следующие функции:

struct matrix read(FILE \*in) — считывает матрицу из файла.

Matrix task – выполняет задание – делит столбец с максимальным значением на максимальное значение

1. В основной части программы будем использовать switch для переключения по меню, в котором есть пять опций:
2. Считываем матрицы из входного файла и выводим сообщение об успешном считывании.
3. Печатаем матрицы A: сначала размер, потом с помощью функции print() выводим матрицу в обычном виде, а затем циклом проходимся по вектору и выводим внутреннее представление матрицы.
4. Ищем максимальный элемент и его столбец, а затем передаем в функцию task
5. Печатаем получившуюся в результате работы task матрицу
6. Завершаем работу программы.

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

Ex1

3x1

0

0

77

Ex2

3x3

0 0 0

2 7 0

0 7 1

Ex3

3x3

1 0 0

0 9 0

1 7 9

Ex4

3x3

1 0 -5

1 0 0

0 0 1

Ex5

5x5

0 0 1 4 0

1 3 6 0 0

1 0 0 0 0

3 0 1 2 0

0 0 3 0 0

Ex6

4x3

1 -8 0 0

0 0 0 0

7 5 0 2

Ex7

5x6

1 0 2 0 -9 0

0 0 3 0 7 9

0 1 0 0 0 0

2 2 0 0 -8 0

0 0 0 0 3 3

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

**8. Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами,подписанныйпреподавателем).

#include <stdio.h>

#include <stdlib.h>

struct matrix{

int V[100];

int N, M;

int not0\_el;

};

typedef struct matrix Matrix;

typedef struct max\_and\_col{

int max;

int col;

}mal;

void menu(){

puts("1. Read matrices A.");

puts("2. Print matrices A.");

puts("3. Task.");

puts("4. Print Task.");

puts("5. Quit.");

}

Matrix read(FILE \*in){

Matrix ans;

int index=0, a;

ans.not0\_el=0;

ans.V[index]=0;

fscanf(in, "%dx%d\n", &ans.N, &ans.M);

for (int i = 1; i <= ans.N; i++) {

index+=1;

ans.V[index]=i;

for (int j = 1; j <= ans.M; j++) {

fscanf(in, "%d", &a);

if (a!=0) {

index+=1;

ans.V[index]=j;

index+=1;

ans.V[index]=a;

ans.not0\_el+=1;

}

}

fscanf(in, "\n");

index+=1;

ans.V[index]=0;

}

index+=1;

ans.V[index]=0;

return ans;

}

void print(Matrix a){

//c - collomn

int index=1;

for (int c=2;c<2\*(a.N+a.not0\_el+1);c+=2){

if (a.V[c]==0){

index=1;

for(int j=a.V[c-2];j<a.M;j++){

printf("%5d",0);

}

printf("\n");

} else {

for(int j=index;j<a.V[c];j++){

printf("%5d",0);

index=a.V[c];

}

printf("%5d",a.V[c+1]);

index+=1;

}

}

}

void print\_v(Matrix a){

for (int i=0; i < 2\*(a.N + a.not0\_el + 1); i++){

printf("%d ", a.V[i]);

}

}

mal col\_max\_el(Matrix a){

mal m;

int max=0;

int collomn, sign = 1;

for (int c=2;c<2\*(a.N+a.not0\_el+1);c+=2){

if ((a.V[c]!=0)&&(abs(a.V[c+1]) > max)){

max = abs(a.V[c+1]);

if (a.V[c+1] < 0){

sign = -1;

}

collomn = a.V[c];

}

}

m.max = max\*sign;

m.col = collomn;

return m;

}

Matrix task(Matrix a, int max\_col, int max){

for (int c=2;c<2\*(a.N+a.not0\_el+1);c+=2){

if ((a.V[c]==max\_col)&&(a.V[c]!=0)){

a.V[c+1]=a.V[c+1]/max;

}

}

return a;

}

int main(){

Matrix A;

int a, g = 1;

FILE \*finput = fopen("exemple7.txt", "r");

if (!finput){

perror("Can't open file");

return 1;

}

A = read(finput);

mal m = col\_max\_el(A);

int max = m.max;

int col = m.col;

menu();

print(A);

printf("%d\n", max );

A = task(A, col, max);

print(A);\*/

while(g){

printf("Enter what do you want ");

scanf("%d", &a);

switch(a) {

case 1:{

printf("The matrices were successfully read\n");

break;

}

case 2: {

printf("Matrix A:\n");

print(A);

printf("Inner data of matrix: ");

print\_v(A);

printf("\n");

break;

}

case 3: {

task(A, col, max);

printf("Task successfully done");

break;

}

case 4: {

A = task(A, col, max);

print(A);

break;

}

case 5:{

printf("The program is closed\n");

return 0;

break;

case 6:

g = 0;

break;

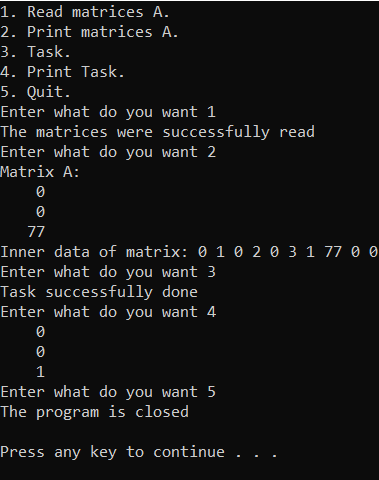
}

}

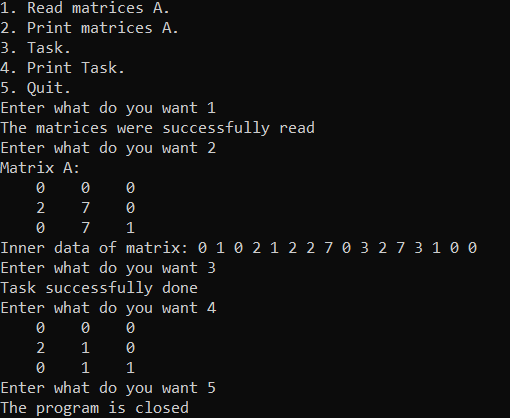
}

}

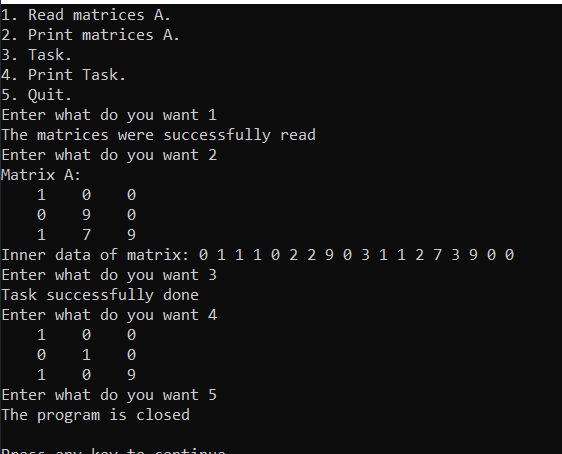
Exemple1:



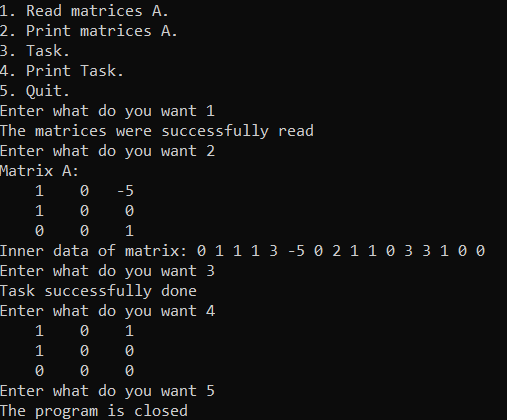
Exemple 2:



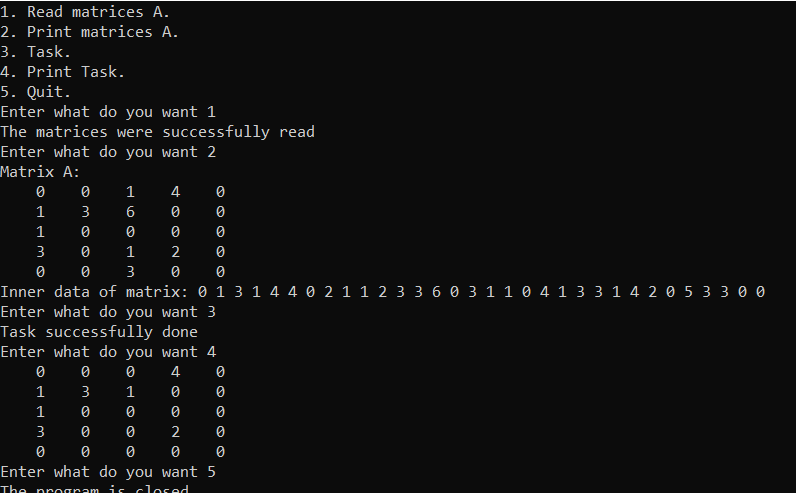
Exemple 3:



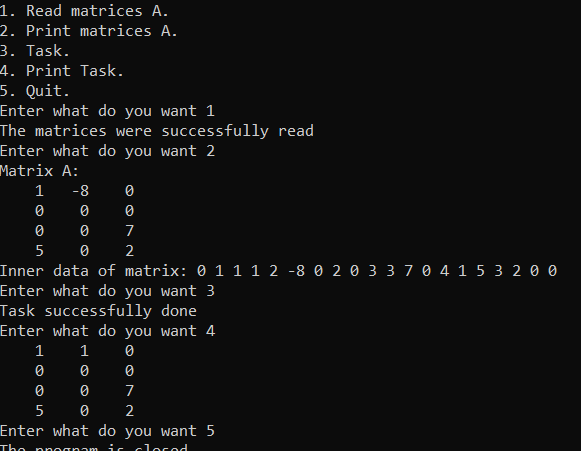
Exemple 4:



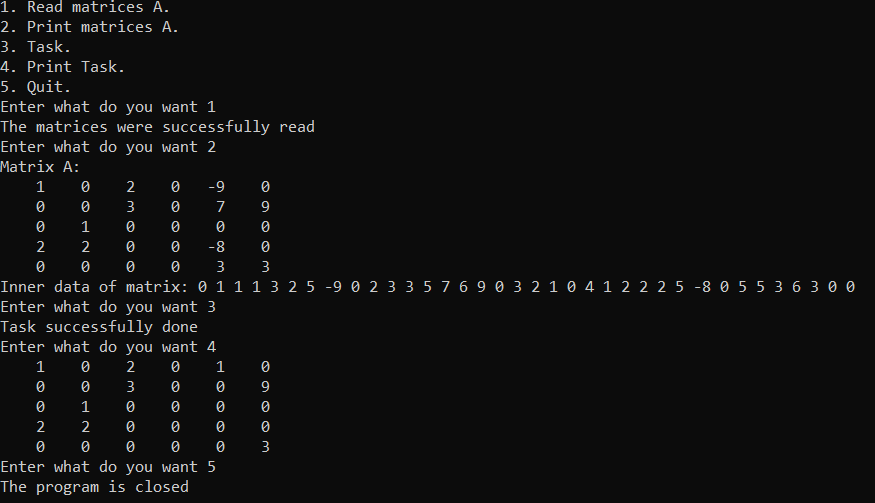
Exemple 5:



Exemple 6:



Exemple 7:



**9. 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. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  | или |  |  |  |  |  |
|  | дом. |  |  |  |  |  |
|  |  |  |  |  |  |  |

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

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

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

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

1. **Выводы**

Я научилась обрабатывать разреженные матрицы на Си

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

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

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