|  |  |
| --- | --- |
| + | **Отчёт по лабораторной работе** №VII  по курсу 1 фундаментальная информатика  студента группы М8О-105Б-21 Бондаревой Елены Евгеньевны, № по списку 1  Контакты www, e-mail, icq, skype : lena\_bondareva\_03@mail.ru  Работа выполнена: «19» апреля 2022г.  Преподаватель: каф.806 В.К.Титов  Входной контроль знаний с оценкой  Отчёт сдан «26» апреля 2022 , итоговая оценка  Подпись преподавателя |

**1. Тема**: «Разреженные матрицы»

**2. Цель работы**: составить программу на языке Си с процедурами или функциями для обработки разреженных матриц, которая выводит матрицы в обычном формате, печатает введенные матрицы во внутр. представлении (2 вида), выполняет заданное действие, а также печатает результат преобразования согласно заданной схеме размещения и в обычном виде.

**3. Задание** (вариант: 1*)*

Схема размещения: цепочка ненулевых элементов в векторе А со строчным индексированием.

Вариант преобразования: определить максимальный по модулю элемент матрицы и разделить на него все элементы строки, в которой он находится. Если таких элементов несколько, обработать каждую строку, содержащую такой элемент.

**4. Оборудование** (*лабораторное*):

ЭВМ -, процессор -, имя узла сети-с ОП -МБ

НМД -ГБ. Терминал - адрес -. Принтер -.

Другие устройства -.

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz , ОП 6 ГБ, НМД 240 ГБ. Монитор IPS 1920x1080

Другие устройства -.

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

Операционная система семейства -, наименование - версия -

Интерпретатор команд - версия -

Система программирования -версия -

Редактор текстов - версия -

Утилиты операционной системы -

Прикладные системы и программы -

Местонахождения и имена файлов программ и данных-

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства UNIX/GNU , наименование Ubuntu версия x86\_64

Интерпретатор команд bash

Редактор текстов emax

Утилиты операционной системы cat

Прикладные системы и программы VTM-diagram

Местонахождения и имена файлов программ и данных -

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

На вход поступает матрица различного размера, представленная во входном текстовом файле в обычном формате(по строкам), с одновременным размещением ненулевых элементов в разреженной матрице. Далее происходит печать введенной матрицы во внутреннем представлении согласно заданной схеме размещения и в обычном(естественном)виде. Затем выполняются необходимые преобразования разреженных матриц:

1) создание (основной) функции нахождения максимального элемента в матрице.

2) создание (основной) функции преобразования матрицы, а именно разделить на найденный максимальный элемент все элементы тех строк, в которых он присутствует.

После чего осуществляется печать результата преобразования согласно заданной схеме размещения и в обычном виде.

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

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <math.h>

#include <locale.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

void print\_short\_matrix(int\* ind, int n, float\* a, int p) {

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

printf("%d ", ind[i]);

printf("\n");

for (int i = 0; i < p; i++)

{

if (i % 2 == 0)

printf("%.0f ", a[i]);

else

printf("%.2f ", a[i]);

}

printf("\n");

}

void print\_full\_matrix(float\*\* matrix, int n, int m) {

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

{

for (int j = 0; j < m; j++)

printf("%.2f\t", matrix[i][j]);

printf("\n");

}

printf("\n");

}

float find\_max(float\* a, int p) {

float max = 0;

for (int i = 1; i < p; i += 2)

{

if (fabs(a[i]) > max)

max = fabs(a[i]);

}

return max;

}

int get\_finish\_row(int i, int\* ind, int n, int p) {

int finish;

if (i == n - 1)

finish = p;

else {

int t = i + 1;

while (t < n && ind[t] == -1)

t++;

if (t == n)

finish = p;

else

finish = ind[t];

}

return finish;

}

void calc\_row\_max(int\* ind, int n, float\* a, int p) {

float max = find\_max(a, p);

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

{

if (ind[i] != -1) {

int start = ind[i];

int finish = get\_finish\_row(i, ind, n, p);

int find = 0;

for (int j = start + 1; j < finish; j += 2)

{

if (fabs(a[j]) == max) {

find = 1;

break;

}

}

if (find == 1) {

for (int j = start + 1; j < finish; j += 2)

a[j] = a[j] / max;

}

}

}

}

float\*\* alloc\_memory(int n, int m) {

float\*\* matrix = (float\*\*)malloc(n \* sizeof(float\*));

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

matrix[i] = (float\*)malloc(m \* sizeof(float));

return matrix;

}

void free\_memory(float\*\* a, int n) {

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

free(a[i]);

free(a);

}

void short\_to\_full(int\* ind, int n, float\* a, int p, float\*\* matrix, int m) {

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

{

for (int j = 0; j < m; j++)

matrix[i][j] = 0;

}

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

{

if (ind[i] != -1) {

int start = ind[i];

int finish = get\_finish\_row(i, ind, n, p);

for (int k = start; k < finish; k += 2)

{

int j = (int)a[k];

matrix[i][j] = a[k + 1];

}

}

}

}

int main()

{

setlocale(LC\_ALL, "rus");

FILE\* file = fopen("in.txt", "r");

int n, m;

fscanf(file, "%d %d", &n, &m);

int\* ind = (int\*)malloc(sizeof(int) \* n);

float\* a = (float\*)malloc(2 \* n \* m \* sizeof(float));

float k;

int p = 0;

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

ind[i] = -1;

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

fscanf(file, "%f", &k);

if (k != 0) {

if (ind[i] == -1)

ind[i] = p;

a[p] = j;

a[p + 1] = k;

p += 2;

}

}

}

fclose(file);

print\_short\_matrix(ind, n, a, p);

calc\_row\_max(ind, n, a, p);

float max = find\_max(a, p);

printf("Максимальный элемент матрицы: %.2f", max);

printf("\n");

printf("После преобразования\n");

print\_short\_matrix(ind, n, a, p);

float\*\* matrix = alloc\_memory(n, m);

short\_to\_full(ind, n, a, p, matrix, m);

printf("В обычном виде\n");

print\_full\_matrix(matrix, n, m);

free(a);

free(ind);

}

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

Допущен к выполнению работы. Подпись преподавателя

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

elena@elena-Aspire-A315-53G:~$ cat tit.txt

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~ ~

~ Лабораторная работа №VII ~

~ Разреженные матрицы ~

~ ~

~ Бондарева Елена ~

~ М8О-105Б-21 ~

~ ~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

elena@elena-Aspire-A315-53G:~$ cat 7.cpp

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <math.h>

#include <locale.h>

#define \_CRT\_SECURE\_NO\_WARNINGS

void print\_short\_matrix(int\* ind, int n, float\* a, int p) {

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

printf("%d ", ind[i]);

printf("\n");

for (int i = 0; i < p; i++)

{

if (i % 2 == 0)

printf("%.0f ", a[i]);

else

printf("%.2f ", a[i]);

}

printf("\n");

}

void print\_full\_matrix(float\*\* matrix, int n, int m) {

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

{

for (int j = 0; j < m; j++)

printf("%.2f\t", matrix[i][j]);

printf("\n");

}

printf("\n");

}

float find\_max(float\* a, int p) {

float max = 0;

for (int i = 1; i < p; i += 2)

{

if (fabs(a[i]) > max)

max = fabs(a[i]);

}

return max;

}

int get\_finish\_row(int i, int\* ind, int n, int p) {

int finish;

if (i == n - 1)

finish = p;

else {

int t = i + 1;

while (t < n && ind[t] == -1)

t++;

if (t == n)

finish = p;

else

finish = ind[t];

}

return finish;

}

void calc\_row\_max(int\* ind, int n, float\* a, int p) {

float max = find\_max(a, p);

printf("Максимальный элемент матрицы: %.2f", max);

printf("\n");

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

{

if (ind[i] != -1) {

int start = ind[i];

int finish = get\_finish\_row(i, ind, n, p);

int find = 0;

for (int j = start + 1; j < finish; j += 2)

{

if (fabs(a[j]) == max) {

find = 1;

break;

}

}

if (find == 1) {

for (int j = start + 1; j < finish; j += 2)

a[j] = a[j] / max;

}

}

}

}

float\*\* alloc\_memory(int n, int m) {

float\*\* matrix = (float\*\*)malloc(n \* sizeof(float\*));

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

matrix[i] = (float\*)malloc(m \* sizeof(float));

return matrix;

}

void free\_memory(float\*\* a, int n) {

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

free(a[i]);

free(a);

}

void short\_to\_full(int\* ind, int n, float\* a, int p, float\*\* matrix, int m) {

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

{

for (int j = 0; j < m; j++)

matrix[i][j] = 0;

}

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

{

if (ind[i] != -1) {

int start = ind[i];

int finish = get\_finish\_row(i, ind, n, p);

for (int k = start; k < finish; k += 2)

{

int j = (int)a[k];

matrix[i][j] = a[k + 1];

}

}

}

}

int main()

{

setlocale(LC\_ALL, "rus");

FILE\* file = fopen("in.txt", "r");

int n, m;

fscanf(file, "%d %d", &n, &m);

int\* ind = (int\*)malloc(sizeof(int) \* n);

float\* a = (float\*)malloc(2 \* n \* m \* sizeof(float));

float k;

int p = 0;

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

ind[i] = -1;

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

fscanf(file, "%f", &k);

if (k != 0) {

if (ind[i] == -1)

ind[i] = p;

a[p] = j;

a[p + 1] = k;

p += 2;

}

}

}

fclose(file);

print\_short\_matrix(ind, n, a, p);

calc\_row\_max(ind, n, a, p);

float max = find\_max(a, p);

printf("После преобразования\n");

print\_short\_matrix(ind, n, a, p);

float\*\* matrix = alloc\_memory(n, m);

short\_to\_full(ind, n, a, p, matrix, m);

printf("В обычном виде\n");

print\_full\_matrix(matrix, n, m);

free(a);

free(ind);

}

elena@elena-Aspire-A315-53G:~$ cat in.txt

4 5

0 3 20 0 7

1 0 3 0 0

0 0 0 0 0

0 0 11 0 20

elena@elena-Aspire-A315-53G:~$ g++ 7.cpp

elena@elena-Aspire-A315-53G:~$ ./a.out in.txt

0 6 -1 10

1 3.00 2 20.00 4 7.00 0 1.00 2 3.00 2 11.00 4 20.00

Максимальный элемент матрицы: 20.00

После преобразования

0 6 -1 10

1 0.15 2 1.00 4 0.35 0 1.00 2 3.00 2 0.55 4 1.00

В обычном виде

0.00 0.15 1.00 0.00 0.35

1.00 0.00 3.00 0.00 0.00

0.00 0.00 0.00 0.00 0.00

0.00 0.00 0.55 0.00 1.00

elena@elena-Aspire-A315-53G:~$ cat > in.txt

5 6

0 0 3 -11 7

1 0 0 5 0

0 -2 3 0 10

11 0 0 3 1

0 0 0 0 2

4 3 0 -1 0

elena@elena-Aspire-A315-53G:~$ ./a.out in.txt

0 8 12 18 22

2 3.00 3 -11.00 4 7.00 5 1.00 2 5.00 5 -2.00 0 3.00 2 10.00 3 11.00 0 3.00 1 1.00 0 2.00 1 4.00 2 3.00 4 -1.00

Максимальный элемент матрицы: 11.00

После преобразования

0 8 12 18 22

2 0.27 3 -1.00 4 0.64 5 0.09 2 5.00 5 -2.00 0 0.27 2 0.91 3 1.00 0 3.00 1 1.00 0 2.00 1 4.00 2 3.00 4 -1.00

В обычном виде

0.00 0.00 0.27 -1.00 0.64 0.09

0.00 0.00 5.00 0.00 0.00 -2.00

0.27 0.00 0.91 1.00 0.00 0.00

3.00 1.00 0.00 0.00 0.00 0.00

2.00 4.00 3.00 0.00 -1.00 0.00

**9. Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) 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 | дом | 19.04.  2022 | 19:55 | В функции short\_to\_full поставила лишнюю скобку | Внимательно писать код. |  |

**10**. Замечание автора по существу работы

**11.** Выводы

В результате выполнения работы я составила программу на языке Си для обработки разреженных матриц, которая выводит матрицы в обычном формате, печатает введенные матрицы во внутр. представлении (2 вида), выполняет заданное действие, печатает результат преобразования в 2 видах.

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом

Подпись студента