Отчёт по лабораторной работе № VII

по курсу 1 Практикум на ЭВМ

студента группы М8О-102Б-21 Кажекин Денис,

№ по списку 9

Адреса www, e-mail, jabber, skype deniskazhekin@mail.ru

Работа выполнена: "____" _____ 20____г.

Преподаватель: доцент каф. 806 Никулин С.П.

Входной контроль знаний с оценкой _____

Отчёт сдан "____" _____ 20____г., итоговая оценка _____

Подпись 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 (группы 6, 8), вещественного (группы 2-5) или комплексного (группы 1, 7) типов, 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 матрицы во внутреннем представлении согласно заданной схеме 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.

   В процедурах и функциях предусмотреть проверки и печать сообещний в случаях ошибок в задании параметров. Для отладки использовать матрицы, содержащие 5-10% ненулевых элементов с максимальным числом элементов 100. Все матрицы m * n хранятся по строкам, в порядке возрастания индексов ненулевых элементов.

3. **Задание**:

   Вариант физического представления: отображение на массив.

   Вариант преобразования: найти столбец, содержащий наибольшее количество ненулевых элементов, и напечатать его номер и произведение элементов этого столбца. Если таких будет несколько обработать предпоследний.

   Вариант схемы размещения:

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

М:	Индекс начала 1-ой строки в массиве A	Индекс начала 2-й строки	...	Индекс начала N-ой Строки

A:	Номер столбца	Значение	Индекс следующего ненулевого элемента этой строки (или 0)	Номер столбца	Значение	Индекс следующего ненулевого элемента этой строки (или 0)	...

4. **Оборудование** (*лабораторное*):
   Оборудование ПЭВМ студента, если использовалось:
   Процессор: Apple M1, с ОП 8192 Мб, НМД 262144 Мб. Монитор: Retina 13,3; IPS 2560 x 1600.

5. **Программное обеспечение** (*лабораторное*):
   Операционная система семейства Linux, наименование Ubuntu, версия ____20.04.3 LTS____интерпретатор команд _bash_ версия __5.0.17(1)__
   Редактор текстов GNU emacs, версия 27.2
   Прикладные системы и программы
6. **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальное описание с пред- и постусловиями)

Алгоритм программы:

Принимая файл с матрицей, проверяем на ошибки, а затем представляется возможность выбора в меню необходимого действия:

1) Вывод первоначальной матрицы
2) Вывод внутреннего представления матрицы
3) Производим вычисления и выводим их
4) Выход из программы

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

Структура

void Printmatrix2(int* A, double* B, int k, int count) – функция, выводящая матрицу во внутреннем представлении

void Printmatrix(int* A, double* B, int n, int m) – функция, выводящая первоначальную матрицу

int menu() – меню программы

void CalculMatrix(int *columns, double* B, int n, int m, int count) – функция, которая выводит результат вычислений

void without_filename(), void file_not_exist(char* file) – функция ошибки

int main(int argc, char* argv[]) - работает с файлом, которая выполняет работу по заданию.

*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)

```
deniskazhekin@MacBook-Air-Denis ~ % cat main.c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void CalculMatrix(int *columns, double* B, int n, int m, int count){
	int k;
	double peremn = 1;
	for(int i = 0; i < count; i+=3){
		k = B[i] - 1;
		columns[k]++;
	}
	int max1 = 0, max2 = 0, maxst1, maxst2, maxst;
	for(int i = 0; i < n; i++){
		if(columns[i] >= max1){
			max2 = max1;
			maxst2 = maxst1;
			max1 = columns[i];
			maxst1 = i;
		}
		else {
			if(columns[i] >= max2){
				max2 = columns[i];
```

```
                                        maxst2 = i;
                                }
                        }
                }
                printf("\n");
                if(max1 == max2){
                        maxst = maxst2 + 1;
                }
                else{
                        maxst = maxst1 + 1;
                }
                for(int i = 0; i < count; i+=3){
                        if(B[i] == maxst)
                                peremn *= B[i + 1];
                }
                printf("%lf \n", peremn);
}




void Printmatrix2(int* A, double* B, int k, int count){
        printf(" Massiv A: \n");
        for(int i = 0; i < k; i++){
                printf(" %d ", A[i]);
        }
        printf("\n Massiv B: ");
        for(int i = 0; i < count; i++){
                if(i % 3 == 0)
                        printf("\n");
                printf(" %.4lf ", B[i]);
        }
        printf("\n");
}

void Printmatrix(int* A, double* B, int n, int m){
        int count = 0, ost, kolvo = 0, kolvo1 = 0;
        double check = 1;
        for(int i = 0; i < m; i++){
                kolvo = 0;
                if(A[i] == 0){
                        for(int j = 0; j < n; j++){
                                printf("0 \t");
                        }
                        printf("\n");
                        continue;
                }
                else {
                        for(int j = 0; j < B[count] - 1; j++){
                                printf("0 \t");
                                kolvo++;
                        }
                        printf("%.4lf \t", B[count + 1]);
                        kolvo++;
                        if(B[count + 2] == 0){
                                ost = n - B[count];
                                for(int j = 0; j < ost; j++){
                                        printf("0 \t");
                                }
                        }
                        else{
                                check = B[count + 5];
                                while(check != 0){
                                        count += 3;
                                        kolvo1 = kolvo;
                                        for(int j = kolvo1; j < B[count] -1; j++){
                                                printf("0 ");
                                                kolvo++;
                                        }
                                        printf("%.4lf \t", B[count + 1]);
                                        kolvo++;
                                        check = B[count + 2];
```

```
                                        }
                                        for(int j = kolvo; j < n; j++)
                                                printf("0 ");
                                }
                        }
                        count += 3;
                        printf("\n");
                }
}

int menu() {
   printf("%s\n", "1. Вывод первоначальной матрицы");
   printf("%s\n", "2. Вывод внутреннего представления матрицы");
   printf("%s\n", "3. Вывод результат после вычислений");
   printf("%s\n", "4. Выход");
   int l;
   scanf("%d", &l);
   return l;
}

void without_filename()
{
        printf("Error.\nUsage: FILE.\n");
}

void file_not_exist(char* file)
{
        printf("File \"%s\" not exist.\nUsage: FILE.\n",file);
}

int main(int argc, char* argv[]) {
        int m, n, count = 0, nachalo_stroki = 0, i1 = -1, k = 0, massiv = 3, cheker = 0;
        int *A;
        int *columns;
        double *B;
        B = (double*)malloc(massiv * sizeof(double));

        if(argc<1) {
                without_filename();
                return 1;
      }
        FILE* file;
      file = 0;
      file  = fopen(argv[1],"r");
      if(!file) {
                file_not_exist(argv[1]);
                return 2;
      }

        fscanf(file,"%d %d", &m, &n);
        A = (int*)calloc(m, sizeof(int));
        columns = (int*)calloc(n, sizeof(int));
        for (int i = 0; i < m; i++){
                nachalo_stroki = 0;
                for (int j = 0; j < n; j++){
                double element;
                fscanf(file, "%lf", &element);
                if (element != 0) {
                        if(nachalo_stroki == 0){
                                nachalo_stroki = 1;
                                A[k] = count + 1;
                                k++;

                        }

                        if(cheker == 0){
                                cheker = 1;
                        }
                        else {
                                massiv += 3;
                                B = (double*)realloc(B, massiv * 10 * sizeof(double));
                        }
```

```
                        B[count] = j + 1;
                        B[count + 1] = element;
                        if(i1 == i){
                                B[count - 1] = count + 1;
                        }
                        else{
                                i1 = i;
                        }
                        count += 3;
                }
            }
            if(nachalo_stroki == 0){
                A[k] = 0;
                k++;
            }
        }
        B[count - 1] = 0;

        int l = 0;
        while (l != 4) {
            l = menu();
            switch (l)
            {
            case 1:
            {
               Printmatrix(A, B, n, m);
               break;
            }
            case 2:
            {
               Printmatrix2(A, B, k, count);
               break;
            }
            case 3:
            {
                CalculMatrix(columns, B, n, m, count);
               break;
            }
            case 4:
            {
               break;
            }
            default:
               printf("%s\n", "Try again");
            }
            }

            free(B);
            free(A);
            fclose(file);
            return 0;
}
deniskazhekin@MacBook-Air-Denis ~ % cat > file
1.0000 0 0 0 0
1.0000 0 0 0 0
1.0000 0 0 0 0
1.0000 0 0 0 0
1.0000 0 0 0 0
deniskazhekin@MacBook-Air-Denis ~ % gcc main.c -o a.out
deniskazhekin@MacBook-Air-Denis ~ % ./a.out file
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход



1
1.0000 0 0 0 0
1.0000 0 0 0 0
```

```
1.0000 0 0 0 0
1.0000 0 0 0 0
1.0000 0 0 0 0

1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
3
1.000000
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
2
Massiv A:
 1  4  7  10  13
 Massiv B:
1.0000  1.0000  0.0000
1.0000  1.0000  0.0000
1.0000  1.0000  0.0000
1.0000  1.0000  0.0000
1.0000  1.0000  0.0000

1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
4
deniskazhekin@MacBook-Air-Denis ~ % cat > file
1.0000   0     0          0          0
0        0     0          0          0
0        0     0          0          0
0        0     0          5.0000     0
0        0     0          10.0000    0
deniskazhekin@MacBook-Air-Denis ~ % gcc main.c -o a.out
deniskazhekin@MacBook-Air-Denis ~ % ./a.out fil
File "fil" not exist.
Usage: FILE.
deniskazhekin@MacBook-Air-Denis ~ % ./a.out file
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
1
1.0000   0     0          0          0
0        0     0          0          0
0        0     0          0          0
0        0     0          5.0000     0
0        0     0          10.0000    0
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
2
 Massiv A:
 1  0  0  13  16
 Massiv B:
 1.0000  1.0000  0.0000
 4.0000  5.0000  0.0000
 4.0000  10.0000  0.0000
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
3
50.000000
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
4
```

```
deniskazhekin@MacBook-Air-Denis ~ % cat > file
1.0000   0      0          0          0
0        0      0          0          0
0        0      3.0000     2.0000     6.0000
0        0      0          0          0
0        0      0          0          0
deniskazhekin@MacBook-Air-Denis ~ %./a.out file
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
1
1.0000   0      0          0          0
0        0      0          0          0
0        0      3.0000     2.0000     6.0000
0        0      0          0          0
0        0      0          0          0
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
2
 Massiv A:
 1  0  4  0  0
 Massiv B:
 1.0000  1.0000  0.0000
 3.0000  3.0000  7.0000
 4.0000  2.0000  10.0000
 5.0000  6.0000  0.0000
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
3
2.000000
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
4
deniskazhekin@MacBook-Air-Denis ~ % cat > file
0          50.0000    0          0          0
0          0          0          0          20.0000
0          0          0          8.0000     0
0          0          0          0          4.0000
55.0000    0          0          0          0
deniskazhekin@MacBook-Air-Denis ~ %./a.out file
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
1
0          50.0000    0          0          0
0          0          0          0          20.0000
0          0          0          8.0000     0
0          0          0          0          4.0000
55.0000    0          0          0          0
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
2
 Massiv A:
 1  4  7  13  16


Massiv B:
 2.0000  50.0000  0.0000
 5.0000  20.0000  0.0000
 4.0000  8.0000  0.0000
 5.0000  4.0000  0.0000
 1.0000  55.0000  0.0000
```

1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
3
80.000000
1. Вывод первоначальной матрицы
2. Вывод внутреннего представления матрицы
3. Вывод результат после вычислений
4. Выход
4

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

10. Замечание автора по существу работы _____

11. Выводы Я научился работать с разреженными матрицами

Подпись студента _____