Министерство образования и науки Украины
Севастопольский национальный технический университет

ПРОГРАММИРОВАНИЕ ОПЕРАЦИЙ НАД МАТРИЦАМИ

Методические указания

к выполнению лабораторной работы № 7
по дисциплине
«Алгоритмические языки и программирование»
для студентов направления подготовки
0914 «Компьютеризированные системы, автоматика и управление»
дневной формы обучения

Севастополь
2006

Create PDF files without this message by purchasing novaPDF printer (http://www.novapdf.com)

СОДЕРЖАНИЕ

Create PDF files without this message by purchasing novaPDF printer (http://www.novapdf.com)

1. ЛАБОРАТОРНАЯ РАБОТА №7
ПРОГРАММИРОВАНИЕ ОПЕРАЦИЙ НАД МАТРИЦАМИ

1.1. Цель работы

Приобретение навыков алгоритмизации и программирования матричных операций, а также манипулирования внешними файлами.

1.2. Работа с файлами

Исходные значения элементов векторов и матриц удобно хранить на внешних носителях в виде файлов.

В языке Си отсутствуют операторы для работы с файлами. Все необходимые действия выполняются с помощью функций, включенных в стандартную библиотеку. Они позволяют работать с различными устройствами, такими, как диски, принтер, и т.д. Эти устройства сильно отличаются друг от друга. Однако файловая система преобразует их в единое абстрактное логическое устройство, называемое потоком.

В Си существует два типа потоков: текстовые (text) и двоичные (binary).

Текстовый поток - это последовательность символов. При передаче символов из потока на экран, часть из них не выводится (например, символ возврата каретки, перевода строки).

Двоичный поток - это последовательность байтов, которые однозначно соответствуют тому, что находится на внешнем устройстве.

Прежде чем читать или записывать информацию в файл, он должен быть открыт и тем самым связан с потоком. Это можно сделать с помощью библиотечной функции fopen( ). Она берет внешнее представление файла (например, c:\my_prog.txt) и связывает его с внутренним логическим именем, которое используется далее в программе. Логическое имя - это указатель на требуемый файл. Его необходимо определить; делается это, например, так:

```
FILE *fp;
```

Здесь FILE - имя типа, описанное в стандартном заголовочном файле stdio.h, fp - указатель на файл. Обращение к функции fopen( ) в программе осуществляется выражением:

```
fp = fopen(спецификация файла, "способ использования файла");
```

Спецификация файла (т.е. имя файла и путь к нему) может, например, иметь вид: "c:\\my_prog.txt" - для файла my_prog.txt на диске c:.

Способ использования файла задается следующими символами:

r - открыть существующий файл для чтения;

w - создать новый файл для записи (если файл с указанным именем существует, то он будет переписан);

a - дополнить файл (открыть существующий файл для записи информации, начиная с конца файла, или создать файл, если он не существует);

r+ - открыть существующий файл для чтения и записи;

w+ - создать новый файл для чтения и записи;

Create PDF files without this message by purchasing novaPDF printer (http://www.novapdf.com)

a+ - дополнить или создать файл с возможностью чтения и записи;

Если в результате обращения к функции fopen( ) возникает ошибка, то она возвращает указатель на константу NULL.

После окончания работы с файлом он должен быть закрыт. Это делается с помощью библиотечной функции fclose(). Она имеет следующий прототип:

```
int fclose(FILE *fp);
```

При успешном завершении операции функция fclose( ) возвращает значение нуль. Любое другое значение говорит об ошибке.

Рассмотрим другие библиотечные функции, используемые для работы с файлами (все они описаны в файле stdio.h):

1. Функция putc( ) записывает символ в файл и имеет следующий прототип:

```
int putc(int c, FILE *fp);
```

Здесь fp - указатель на файл, возвращенный функцией fopen( ), c - символ для записи (переменная c имеет тип int, но используется только младший байт). При успешном завершении putc( ) возвращает записанный символ, в противном случае возвращается константа EOF. Она определена в файле stdio.h и имеет значение (-1).

2. Функция getc( ) читает символ из файла и имеет следующий прототип:

```
int getc(FILE *fp);
```

Здесь fp - указатель на файл, возвращенный функцией fopen( ). Эта функция возвращает прочитанный символ. Соответствующее значение имеет int, но старший байт равен нулю. Если достигнут конец файла, то getc возвращает значение EOF.

3. Функция feof( ) определяет конец файла при чтении двоичных данных и имеет следующий прототип:

```
int feof(FILE *fp);
```

Здесь fp - указатель на файл, возвращенный функцией fopen( ). При достижении конца файла возвращается ненулевое значение, в противном случае возвращается 0.

4. Функция fputs( ) записывает строку символов в файл. Она отличается от функции puts( ) только тем, что в качестве второго параметра должен быть записан указатель на переменную файлового типа.

Например:

```
fputs("Example", fp);
```

При возникновении ошибки возвращается значение EOF.

5. Функция fgets( ) читает строку символов из файла. Она отличается от функции gets( ) тем, что в качестве второго параметра должно быть записано максимальное число вводимых символов плюс единица, а в качестве третьего - указатель на переменную файлового типа. Строка считывается целиком, если ее длина не превышает указанного числа символов, в противном случае функция возвращает только заданное число символов.

Рассмотрим пример:

```
fgets(string, n, fp);
```

Create PDF files without this message by purchasing novaPDF printer (http://www.novapdf.com)

Функция возвращает указатель на строку string при успешном завершении и константу NULL в случае ошибки либо достижения конца файла.

6. Функция fprintf( ) выполняет те же действия, что и функция printf(), но работает с файлом. Ее отличием является то, что в качестве первого параметра задается указатель на переменную файлового типа.

Например:

fprintf(fp, "%x",a);

7. Функция fscanf( ) выполняет те же действия, что и функция scanf(), но работает с файлом. Ее отличием является то, что в качестве первого параметра задается указатель на переменную файлового типа.

Например:

fscanf(fp, "%x", &a);

При достижении конца файла возвращается значение EOF.

В языке Си открываются стандартные файлы со следующими логическими именами:

stdin - для ввода данных из стандартного входного потока (по умолчанию с клавиатуры);

stdout - для вывода данных в стандартный выходной поток (по умолчанию на экран дисплея);

stderr - файл для вывода сообщений об ошибках (всегда связан с экраном дисплея).

1.3. Пример программы

Ниже приведен пример программы, осуществляющей следующие действия:

- чтение матрицы A из файла matrix.txt,
- вызов функции addConst, прибавляющей заданное значение к каждому элементу квадратной матрицы;
- вывод новой матрицы в файл result.txt.

```
#include <stdio.h>
void addConst(float [][4], float );
int main(void)
{
  const int maxlen=50;
  FILE * f;
  char *s;
  int i,j;
  float a[4][4], num;
  if((f=fopen("matrix.txt","r"))==NULL)
     {
        puts("can't open file");
        return 1;
     }
  fgets(s,maxlen,f);
```

Create PDF files without this message by purchasing novaPDF printer (http://www.novapdf.com)

```
        for (i=0; i<4; i++)
        for (j=0; j<4; j++)
            fscanf(f,"%f", &a[i][j]);
        fclose(f);

        printf("Enter number");
        scanf("%f",&num);
        addConst(a,num);
        if((f=fopen("result.txt","w"))==NULL)
           {
              puts("Can't create file");
              return 1;
           }
        for (i=0; i<4; i++)
            {
              for (j=0; j<4; j++)
                  fprintf(f,"%8.4f", a[i][j]);
                  fputc('\n',f);
            }
        fclose(f);
        return 0;
    }
void addConst(float matr[][4], float c)
{
  int i,j;
  for (i=0; i<4; i++)
  for (j=0; j<4; j++)
      matr[i][j]+=c;
}
```

1.4. Задание на работу

1) Изучите примеры программ, приведенных в разделе 1.2.

2) Получите у преподавателя номер варианта индивидуального задания на лабораторную работу. Варианты заданий приведены в таблице 1.1.

3) Программа должна удовлетворять следующим требованиям:

– в лабораторной работе необходимо реализовать две функции, реализующие вычисления, в соответствии с номером варианта, указанным в таблице 1.1;

– в основной программе необходимо ввести значения элементов заданной матрицы с клавиатуры, а другой матрицы – из текстового файла `matrix.txt`. Содержимое файла `matrix.txt` представлено в приложении А.

– в основной программе вызвать разработанные функции для заданных матриц. Вычислить A*B и B*A или A+B и B+A, A^T и B^T в зависимости от варианта.

Create PDF files without this message by purchasing novaPDF printer (http://www.novapdf.com)

– программа должна осуществлять вывод полученных результатов в отдельный файл.

Таблица 1.1 – Варианты заданий

№ варианта	Задания и значения исходных данных			
	Функция 1	Функция 2	Матрицы	Ввести с клавиатуры
1	Умножение	Максимальная сумма модулей элементов столбца	A, B	A
2	Сложение	Минимальный среди элементов выше главной диагонали	C, D	C
3	Транспонирование	Минимальный среди элементов, расположенных в четных столбцах	E, F	E
4	Умножение	максимальный среди элементов ниже главной диагонали	E, F	E
5	Сложение	Сумма модулей элементов выше главной диагонали	C, D	D
6	Транспонирование	Последний положительный среди элементов ниже главной диагонали	A,B	A
7	Умножение	Корень квадратный из суммы квадратов элементов	C, D	D
8	Сложение	Максимальный среди элементов, расположенных в нечетных столбцах	A, B	A
9	Транспонирование	Минимальный элемент среди положительных	E, F	F
10	Умножение	Первый нулевой среди элементов ниже главной диагонали	C, D	C
11	Сложение	Максимальная сумма модулей элементов строки	E,F	E

Create PDF files without this message by purchasing novaPDF printer (http://www.novapdf.com)

Таблица 1.1 – Варианты заданий

№ варианта	Задания и значения исходных данных			
	Функция 1	Функция 2	Матрицы	Ввести с клавиатуры
12	Транспонирование	Сумма квадратов элементов ниже главной диагонали	C,D	C
13	Умножение	Первый положительный среди элементов ниже главной диагонали	E,F	F
14	Сложение	Максимальный элемент среди отрицательных	C, D	D

1.5. Содержание отчета и порядок защиты работы

Выполнение и защита лабораторной работы производится каждым студентом индивидуально. Защита результатов лабораторной работы осуществляется при наличии работающей программы и полностью оформленного отчета.

Отчет должен включать в себя следующие разделы

- титульный лист;
- цель работы,
- постановка задачи. Этот раздел должен содержать вариант задания в соответствии с таблицей 1.1;
- схема программы. Этот раздел должен содержать схему программы, а также схемы разработанных функций.
- текст программы на языке C;
- результаты работы программы. Этот раздел должен содержать результаты работы программы при значениях матриц, взятых в соответствии с файлом matrix.txt.
- выводы.

Защита работы состоит в следующем:

- представление работающей программы на компьютере;
- предъявление отчета, оформленного в соответствии с указанными требованиями;
- ответы на вопросы преподавателя по теоретической и практической части работы. Примеры возможных вопросов приведены в подразделе 1.5.

1.6. Контрольные вопросы

1. Что собою представляет двумерный массив?
2. Приведите пример фрагмента программы, который бы осуществлял вывод элементов главной диагонали квадратной матрицы на экран.

Create PDF files without this message by purchasing novaPDF printer (http://www.novapdf.com)

3. Дайте определение операции транспонирования матрицы. Приведите пример фрагмента программы, в котором бы осуществлялось транспонирование квадратной матрицы.

4. Дайте определение операции умножения матриц и приведите соответствующую формулу. Как должны быть согласованы размерности матриц? Приведите пример фрагмента программы, в котором бы осуществлялось умножение двух матриц.

5. Что собою представляют файл и связанный с ним поток с точки зрения языка С?

6. Для чего предназначены и как работают функции `fopen` и `fclose`?

7. Для чего предназначены и как работают функции `fprintf` и `printf`?

8. Для чего предназначены и как работают функции `fscanf` и `scanf`?

9. Для чего предназначены и как работают функции `fputs` и `puts`?

10. Для чего предназначены и как работают функции `fgets` и `gets`?

11. Какие переменные называются глобальными, какие – локальными?

12. Что такое формальные параметры? Что такое фактические параметры? Какие соответствия должны соблюдаться между формальными и фактическими параметрами?

13. Каковы особенности передачи параметров-массивов?

Create PDF files without this message by purchasing novaPDF printer (http://www.novapdf.com)

ПРИЛОЖЕНИЕ А

Содержимое файла matrix.txt

```
A =
    1.0668    0.2944   -0.6918   -1.4410
    0        -1.3362    0.8580    0.5711
   -0.0956    0.7143    1.2540    0
   -0.8323    1.6236   -1.5937    0.6900
B =
    0.8156    1.1908   -1.6041   -0.8051
    0.7119   -1.2025    0.2573    0
    1.2902    0        -1.0565    0.2193
    0.6686   -0.1567    1.4151   -0.9219
C =
   -2.1707    0.5077    0.3803    0.0000
   -0.0592    1.6924   -1.0091   0
   -1.0106    0.5913   -0.0195    1.0950
    0.6145   -0.6436   -0.0482   -1.8740
D =
    0.4282    0.0403   -0.3775    0.1184
    0.8956    0.6771   -0.2959    0.3148
    0.7310    0.5689   -1.4751    1.4435
    0.5779   -0.2556   -0.2340   -0.3510
E =
    0.6232    0.2120    1.0823   -0.6355
    0.7990    0.2379   -0.1315   -0.5596
    0.9409   -1.0078    0.3899    0.4437
   -0.9921   -0.7420    0.0880   -0.9499
```

Create PDF files without this message by purchasing novaPDF printer (http://www.novapdf.com)