

ЛАБОРАТОРНАЯ РАБОТА 5.

СОЗДАНИЕ М-ФАЙЛА ДЛЯ ОБРАБОТКИ ВЕКТОРОВ И МАТРИЦ.

5.1. Цель: создание файлов-функций для обработки векторов и матриц.

5.2. Порядок выполнения работы

1. Изучите теоретическую часть. Выполните задания, соответствующие номеру Вашего варианта, и продемонстрируйте их преподавателю (встроенные функции пакета Matlab не использовать).

2. Оформите отчет по лабораторной работе, который должен содержать:

- титульный лист;
- цель работы;
- задание;
- описание программы согласно ГОСТу (общие сведения, функциональное назначение и т.д.)
- листинг программы (m-файл) и результаты выполнения.

5.3. Методические рекомендации

5.3.1. Многооконный редактор/отладчик m-файлов

5.3.1.1. Интерфейс редактора/отладчика m-файлов

Для подготовки, редактирования и отладки m-файлов служит специальный многооконный редактор. Он выполнен как типичное приложение Windows. Редактор можно вызвать командой `edit` из командной строки или командой `New > M-file` из меню `File`. После этого в окне редактора можно создавать свой файл, пользоваться средствами его отладки и запуска. Перед запуском файла его необходимо записать на диск, используя команду `File > Save as` в меню редактора.

На рисунке 1 показано окно редактора/отладчика.

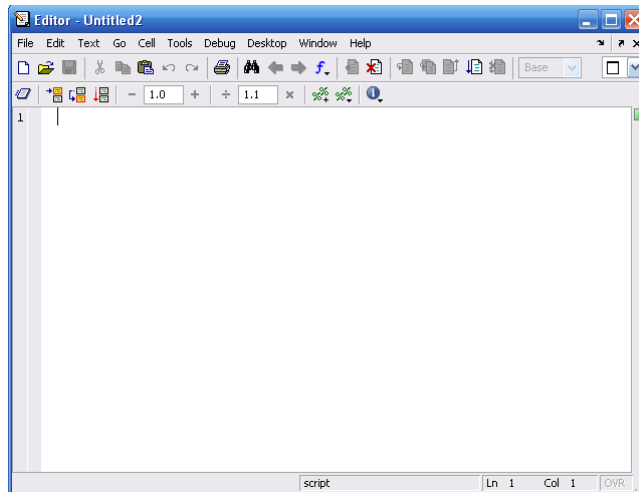


Рисунок 1 – Редактор/отладчик файлов при записи файла на диск

После записи файла на диск можно заметить, что команда `Run` в меню `Debug` редактора становится активной (до записи файла на диск она пассивна) и позволяет произвести запуск файла. Запустив команду `Run`, можно наблюдать исполнение m-файла.

Редактор/отладчик позволяет создать m-файл (программу) без той многочисленной «шелухи», которая сопровождает работу в командном режиме. Текст такого файла подвергается тщательной синтаксической проверке, в ходе которой выявляются и отсеиваются многие ошибки пользователя. Таким образом, редактор обеспечивает синтаксический контроль файла.

Редактор имеет и другие важные отладочные средства – он позволяет устанавливать в тексте файла специальные метки, именуемые точками прерывания (breakpoints). При их достижении вычисления приостанавливаются, и пользователь может оценить промежуточные результаты вычислений (например, значения переменных), проверить правильность выполнения циклов и т. д. Наконец, редактор позволяет записать файл в текстовом формате.

Для удобства работы с редактором/отладчиком строки программы в нем нумеруются в последовательном порядке. Редактор является многооконным. Окно каждой программы оформляется как вкладка.

5.3.1.2. Цветовые выделения и синтаксический контроль

Редактор/отладчик m-файлов выполняет синтаксический контроль программного кода по мере ввода текста. При этом используются следующие цветовые выделения:

- ключевые слова языка программирования – синий цвет;
- операторы, константы и переменные – черный цвет;
- комментарии после знака % – зеленый цвет;
- символьные переменные (в апострофах) – зеленый цвет;
- синтаксические ошибки – красный цвет.

Благодаря цветовым выделениям вероятность синтаксических ошибок снижается.

Однако далеко не все ошибки диагностируются. Ошибки, связанные с неверным применением операторов или функций (например, применение оператора – вместо + или функции $\cos(x)$ вместо $\sin(x)$ и т. д.), не способна обнаружить ни одна система программирования. Устранение такого рода ошибок (их называют семантическими) – дело пользователя, отлаживающего свои алгоритмы и программы.

5.3.1.3. Панель инструментов редактора и отладчика

Редактор имеет свое меню и свою инструментальную панель. Внешний вид инструментальной панели показан на рисунке 2. По стилю данная панель похожа на панель инструментов окна командного режима работы, но имеет несколько иной набор кнопок.



Рисунок 2 – Панель инструментов редактора/отладчика m-файлов

Назначение кнопок панели инструментов редактора/отладчика следующее:

- New – создание нового m-файла;
- Open – вывод окна загрузки файла;
- Save – запись файла на диск;
- Cut – вырезание выделенного фрагмента и перенос его в буфер;
- Copy – копирование выделенного объекта в буфер;
- Paste – размещение фрагмента из буфера в позиции текстового курсора;
- Undo – отмена предшествующей операции;
- Redo – повтор отмененной операции;
- Print – печать содержимого текущего окна редактора;
- Find text – нахождение указанного текста;
- Show function – показ функции;
- Set/Clear Breakpoint – установка/сброс точки прерывания;
- Clear All Breakpoints – сброс всех точек прерывания;
- Step – выполнение шага трассировки;
- Step In – пошаговая трассировка с заходом в вызываемые m-файлы;
- Step Out – пошаговая трассировка без захода в вызываемые m-файлы;
- Run – запись и сохранение;

– Exit Debug Mode – выход из режима отладки.

5.3.1.4 Работа с точками прерывания

Основным приемом отладки m-файлов является установка в их тексте точек прерывания (Breakpoints). Они устанавливаются (и сбрасываются) с помощью кнопки Set/Clear Breakpoint Сброс всех точек прерывания обеспечивается кнопкой Clear All Breakpoints.

Рассмотрим рисунок 3, на котором в окне редактора/отладчика видна конструкция цикла. Прежде всего, для отладки надо записать программу на диск, а затем установить напротив выражения `plot(x,sin(x).^3)` точку прерывания – она отчетливо видна на рисунке 3 как красный кружок. Для установки точки прерывания необходимо поместить текстовый курсор в нужное место (напротив указанного выражения) и щелкнуть на кнопке Set/Clear Breakpoint или щелкнуть справа от номера строки.

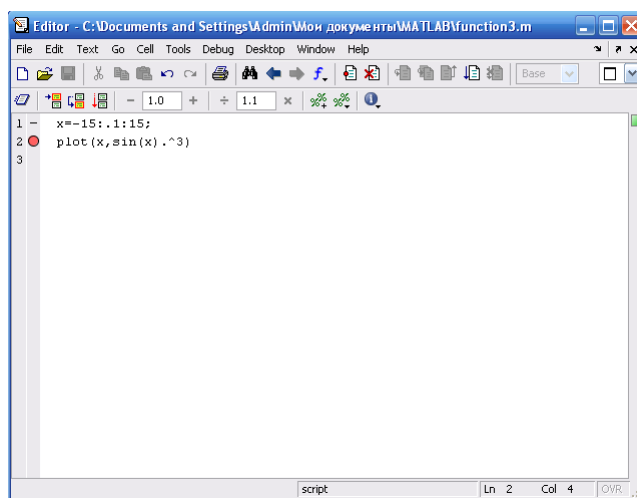


Рисунок 3 – Простейший пример на применение точки прерывания в программе

Теперь при пуске программы командой Run она будет исполнена до точки прерывания, после чего текущие значения будут выведены в окне MATLAB. С помощью кнопки Step (Шаг) можно выполнить очередной шаг вычислений и т. д. Если отпала необходимость останова в точках прерывания, достаточно кнопкой Clear All Breakpoints удалить разом все точки прерывания. Желтая стрелка указывает, в каком месте программы произошла остановка. Обратите внимание на то, что в этом примере каждый шаг исполнения цикла фиксируется в окне командного режима системы MATLAB. При остановке в точке прерывания вы можете провести контроль значений переменных как «вручную», так и с помощью организации вывода на просмотр перед точкой прерывания.

Вы можете задать выполнение программы без остановки при заходе, но с остановкой при выходе (кнопка Step Out), и с остановкой при заходе в вызываемые m-файлы (кнопка Step In). Кнопка Exit Debug Mode (Выход из режима отладки) прекращает операции отладки.

5.3.2. М-файлы сценариев

5.3.2.1. Структура и свойства файлов сценариев

Работа в командном режиме (сессия) не является программированием. Внешним атрибутом последнего в MATLAB служит задание последовательности действий по программе, записанной в виде m-файла. Для создания m-файлов может использоваться как встроенный редактор, так и любой текстовый редактор, поддерживающий формат ASCII. Подготовленный и записанный на диск m-файл становится частью системы, и его можно вызывать как из командной строки, так и из другого m-файла. Есть два типа m-файлов: файлы-сценарии и файлы-функции. Важно, что в процессе своего создания они проходят синтаксический контроль

с помощью встроенного в систему MATLAB редактора/отладчика m-файлов.

Файл-сценарий, именуемый также Script-файлом, является просто записью серии команд без входных и выходных параметров. Он имеет следующую структуру:

```
%Основной комментарий
%Dополнительный комментарий
Тело файла с любыми выражениями
```

Важны следующие свойства файлов-сценариев:

- они не имеют входных и выходных аргументов;
- работают с данными из рабочей области;
- в процессе выполнения не компилируются;
- представляют собой зафиксированную в виде файла последовательность операций, полностью аналогичную той, что используется в сессии.

Основным комментарием является первая строка текстовых комментариев, а дополнительным – последующие строки. Основной комментарий выводится при выполнении команд `lookfor` и `help имя_каталога`. Полный комментарий выводится при выполнении команды `help имя_файла`. Рассмотрим следующий файл-сценарий:

```
%Plot with color red
%Строит график синусоиды линией красного цвета
% выведенной масштабной сеткой в интервале [xmin, xmax]
x=xmin:0.1:xmax;
plot(x, sin(x), 'r')
grid on
```

Первые три строки здесь – это комментарий, остальные – тело файла. Обратите внимание на возможность задания комментария на русском языке. Знак % в комментариях должен начинаться с первой позиции строки. В противном случае команда `help name` не будет воспринимать комментарий (иногда это может понадобиться) и возвратит сообщение вида `No help comments found in-name.m`.

Обратите внимание на то, что такой файл нельзя запустить без предварительной подготовки, сводящейся к заданию значений переменным `xmin` и `xmax`, использованным в теле файла. Это следствие первого свойства файлов-сценариев – они работают с данными из рабочей области. Переменные, используемые в файлах-сценариях, являются глобальными, т.е. они действуют одинаково в командах сессии и внутри программного блока, которым является файл-сценарий. Поэтому заданные в сессии значения переменных используются и в теле файла. Имена файлов-сценариев нельзя использовать в качестве параметров функций, поскольку файлы-сценарии не возвращают значений. Можно сказать, что файл-сценарий – это простейшая программа на языке программирования MATLAB. Файлы-сценарии нельзя компилировать. Перед компилированием их нужно преобразовать в файлы-функции.

На рисунке 4 приведен результат выполнения файла `psr.m`.

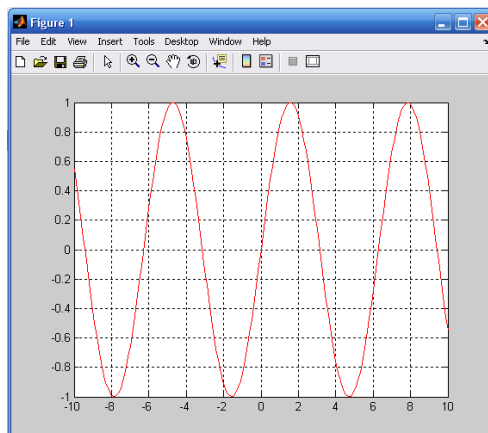


Рисунок 4 – Пример работы с файлом `psf`

5.3.3. М-файлы-функции

5.3.3.1. Структура М-файла-функции

М-файл-функция является типичным объектом языка программирования системы MATLAB. Одновременно он является полноценным модулем с точки зрения структурного программирования, поскольку содержит входные и выходные параметры и использует аппарат локальных переменных. Структура такого модуля с одним выходным параметром выглядит следующим образом:

```
function var=f_name(Список_параметров)
%Основной комментарий
%Дополнительный комментарий
Тело файла с любыми выражениями
var=выражение
```

М-файл-функция имеет следующие свойства:

- он начинается с объявления `function`, после которого указывается имя переменной `var` – выходного параметра, имя самой функции и список ее входных параметров;
- функция возвращает свое значение и может использоваться в виде `name (Список_параметров)` в математических выражениях;
- все переменные, имеющиеся в теле файла-функции, являются *локальными*, т.е. действуют только в пределах тела функции;
- файл-функция является самостоятельным программным модулем, который общается с другими модулями через свои входные и выходные параметры;
- правила вывода комментариев те же, что у файлов-сценариев;
- файл-функция служит средством расширения системы MATLAB;
- при обнаружении файла-функции он компилируется и затем выполняется, а созданные машинные коды хранятся в рабочей области системы MATLAB.

Последняя конструкция `var = выражение` вводится, если требуется, чтобы функция возвращала результат вычислений.

Приведенная форма файла-функции характерна для функции с одним выходным параметром. Если выходных параметров больше, то они указываются в квадратных скобках после слова `function`. При этом структура модуля имеет следующий вид:

```
function [var1,var2,...]=f_name(Список_параметров)
%Основной комментарий
%Dополнительный комментарий
Тело файла с любыми выражениями
var1=выражение
var2=выражение
```

Такая функция во многом напоминает процедуру. Ее нельзя слепо использовать непосредственно в математических выражениях, поскольку она возвращает не единственный результат, а множество результатов – по числу выходных параметров. Если функция используется как имеющая единственный выходной параметр, но имеет ряд выходных параметров, то для возврата значения будет использоваться первый из них. Это зачастую ведет к ошибкам в математических вычислениях. Поэтому, как отмечалось, данная функция используется как отдельный элемент программ вида:

```
[var1,var2,...]=f_name(Список_параметров)
```

После его применения переменные выхода *var1*, *var2*,... становятся определенными и их можно использовать в последующих математических выражениях и иных сегментах программы. Если функция используется в виде *name*(Список_параметров), то возвращается значение только первого выходного параметра – переменной *var1*.

5.3.3.2. Статус переменных в функциях и команда *global*

Переменные, указанные в списке параметров функции, являются *локальными* и служат для переноса значений, которые подставляются на их место при вызовах функций. Эта особенность переменных-параметров хорошо видна при разборе примера, показанного на рисунке 5. Здесь задана некоторая функция двух переменных *fun(x, y)*. В этом примере в окне редактора создана функция *fun* двух переменных *x* и *y*, вычисляющая $z = x^2 + y^2$. Поскольку переменные *x* и *y* указаны как параметры функции *fun(x, y)*, то они являются локальными. В примере вне тела функции им заданы нулевые значения. Очевидно, что при вычислении значения *fun(2, 3)* в теле функции задается $x = 2$ и $y = 3$. Поэтому результат – $z = 13$. Однако, после выхода из тела функции переменные *x* и *y* принимают свои исходные значения, равные нулю. Так что, эти переменные меняют свои значения на значения параметров функции только локально – в пределах тела функции. А каков статус переменной *z* в нашем примере? Она, как и любая переменная, определенная в теле функции, также будет локальной. Изначально ее значение не определено. В теле функции переменная принимает значение $z = 13$. А после возврата из функции, как нетрудно увидеть из рисунка 5, переменная *z*, несмотря на ее применение в теле функции, остается неопределенной. На это указывает сообщение, отображаемое после попытки вывода значения переменной *z*.

Возврат из функции производится после обработки всего тела функции, т.е. при достижении конца файла функции. При использовании в теле функции условных операторов, циклов или переключателей иногда возникает необходимость осуществить возврат функции раньше, чем будет достигнут конец файла. Для этого служит команда *return*. В любом случае, результатом, возвращаемым функцией, являются значения выходных, присвоенные им на момент возврата.

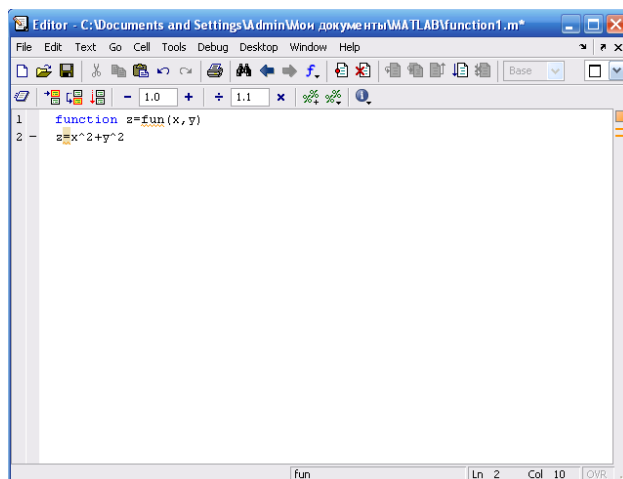


Рисунок 5 – Пример, поясняющий действие локальных и глобальных переменных при задании файла-функции

У нашей функции имеется один недостаток – вывод на индикацию значения $z = 13$ из тела функции, хотя после этого z остается равным 0. Чтобы убрать побочный эффект вывода значения z , достаточно установить знак $;$ после математического выражения, определяющего z . Таким образом, окончательно наша функция должна записываться следующим образом:

```
function z=fun(x,y)
z=x^2+y^2;
```

Этот пример наглядно показывает, что пропуск любого слова или даже простого оператора (вроде знака $;$) может привести к не сразу понятным побочным эффектам и даже неверной работе функции. Программирование требует особой точности и педантичности, именно поэтому далеко не все могут быть хорошими программистами.

Итак, из сказанного ясно, что переменные в файлах-сценариях являются *глобальными*, а в файлах-функциях – *локальными*. Нередко применение глобальных переменных в программных модулях может приводить к побочным эффектам. Применение локальных переменных устраняет эту возможность и отвечает требованиям структурного программирования.

Однако передача данных из модуля в модуль в этом случае происходит только через входные и выходные параметры, что требует тщательного планирования такой передачи. При создании файлов-функций порой желательно применение глобальных переменных. Ответственность за это должен брать на себя программист, создающий программные модули.

Команда `global var1 var2...` позволяет объявить переменные модуля-функции глобальными. Таким образом, внутри функции могут использоваться и такие переменные, если это нужно по условиям решения вашей задачи.

5.3.4. Использование подфункций

Начиная с версии 5.0 в функции системы MATLAB можно включать *подфункции*. Они объявляются и записываются в теле основных функций и имеют идентичную им конструкцию. Не следует путать эти функции с внутренними функциями, встроенными в ядро системы MATLAB. Ниже представлен пример функции с подфункцией:

```
function [mean, stdev] = stat(x)
%Пример функции с встроенной подфункций
n = length(x);
mean = avg(x, n);
```

```
stdev = sqrt(sum((x-avg(x,n)).^2)/n);
```

```
%-----
```

```
function m = avg(x, n)
```

```
m = sum(x)/n;
```

В этом примере среднее значение элементов вектора x вычисляется с помощью подфункции `avg(x, n)`, тело которой записано в теле основной функции `stat`. Пример использования функции `stat` представлен ниже:

```
>> V=[1 2 3 4 5]
```

```
V=
```

```
1 2 3 4 5
```

```
>> [a, m]=function2(V)
```

```
a =
```

```
3
```

```
m =
```

```
1.4142
```

Подфункции определены и действуют локально, т.е. только в пределах `m`-файла, определяющего основную функцию. Заданные в некотором `m`-файле подфункции нельзя использовать ни в командном режиме работы, ни в других `m`-файлах. При обращении к функции интерпретатор системы MATLAB, прежде всего, просматривает `m`-файл на предмет выявления подфункций. Если они обнаружены, то задаются как локальные функции. Благодаря локальному действию подфункций их имена могут совпадать с именами основных функций системы. Если в функции и подфункциях должны использоваться общие переменные, их надо объявить глобальными как в функции, так и в ее подфункциях.

5.3.5. Примеры `m`-файлов для обработки векторов и матриц

Рассмотрим создание `m`-файлов для обработки векторов и матриц.

Контрольный пример 1. Разработать `m`-файл для решения следующей задачи. Если у массива $A(30)$ есть элемент, равный B , то переменной X присвоить значение, равное сумме всех положительных четных элементов, предшествующих первому по порядку такому элементу, иначе переменной X присвоить 0.

Решение. Созданный `m`-файл представлен на рисунке 6. Результат выполнения программы приведен на рисунке 7.

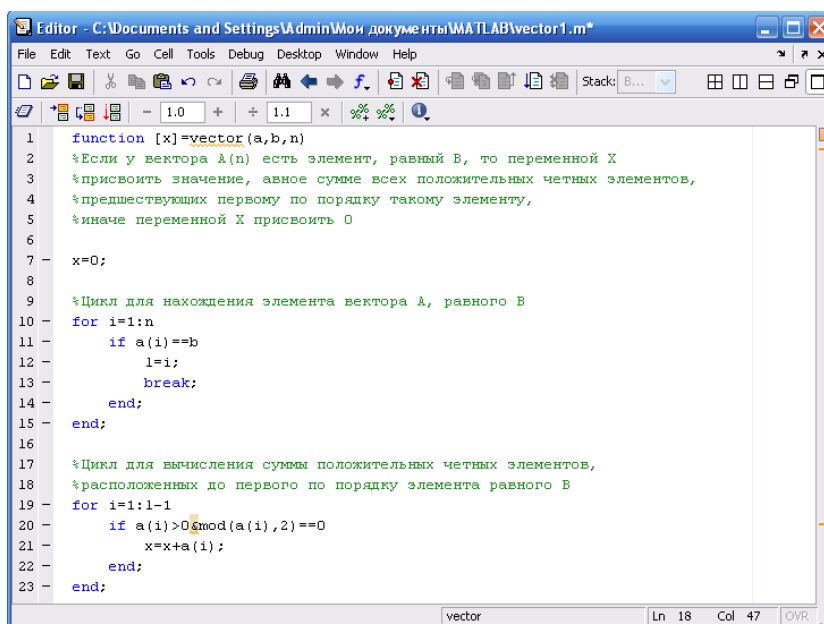


Рисунок 6 – Разработанный `m`-файл `vector1.m` для решения задачи на матрицы


```
Command Window
>> a=[3 2 -2 3 4 -4 -3 6 7 6]

a =

    3     2    -2     3     4    -4    -3     6     7     6

>> b=6

b =

     6

>> vector1(a,b,10)

ans =

     6
```

Рисунок 7 – Результат выполнения m-файла vector1.m

Контрольный пример 2. Разработать m-файл для решения следующей задачи. Даны две действительные квадратные матрицы порядка n . Получить новую матрицу умножением элементов каждой строки первой матрицы на наибольшее из значений элементов соответствующей строки второй матрицы.

Решение. Созданный m-файл представлен на рисунке 8. Результат выполнения программы приведен на рисунке 9.

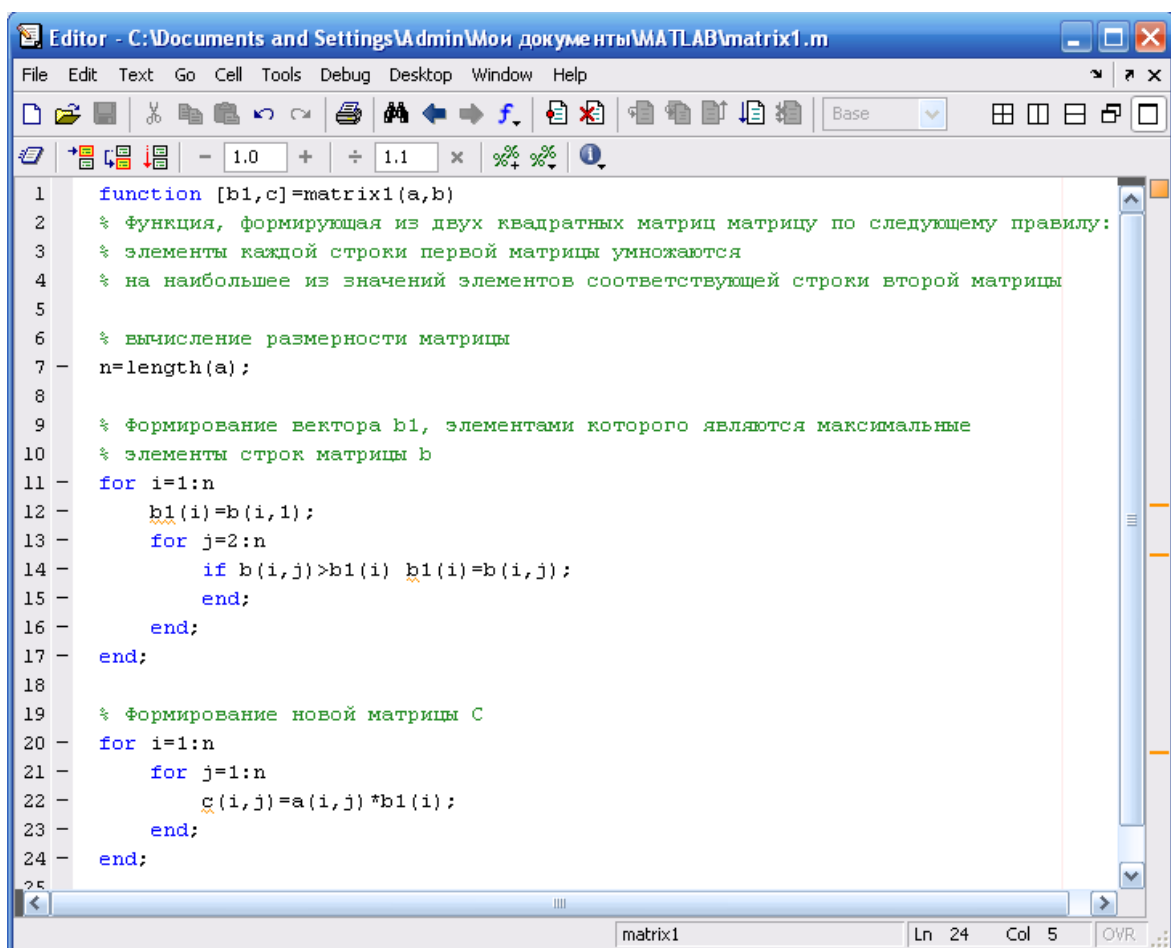
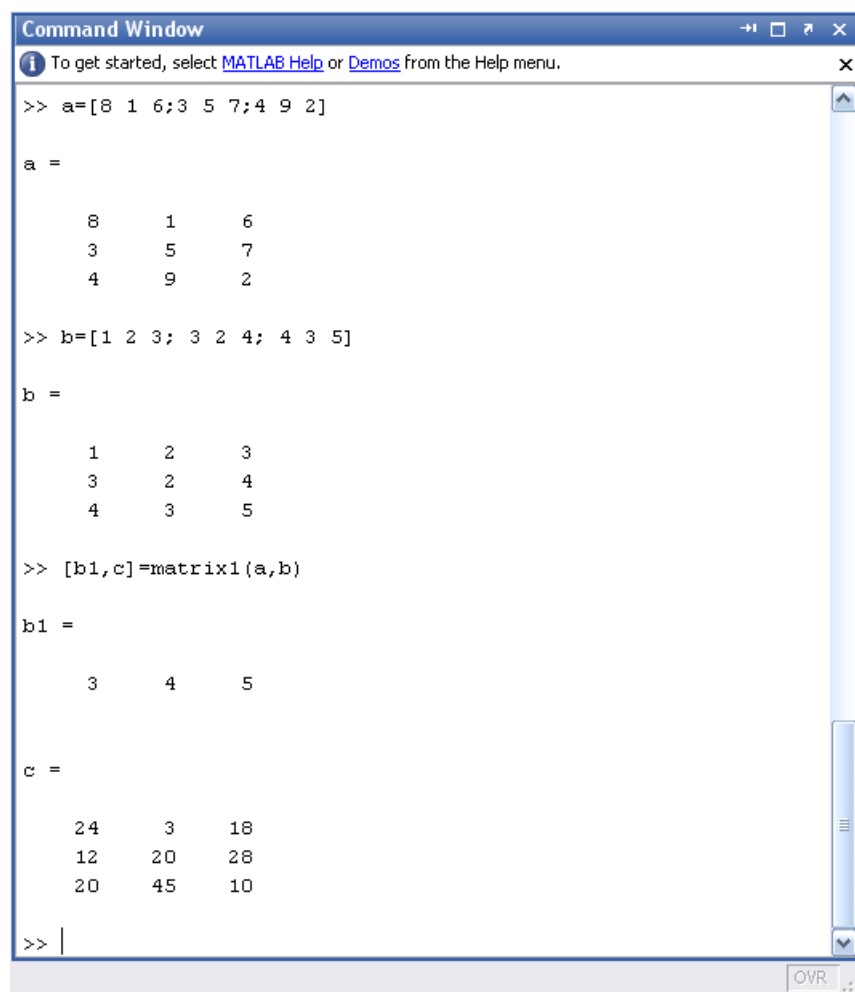


Рисунок 8 – Разработанный m-файл matrix1.m для решения задачи на матрицы



```
Command Window
1 To get started, select MATLAB Help or Demos from the Help menu.

>> a=[8 1 6;3 5 7;4 9 2]

a =

     8     1     6
     3     5     7
     4     9     2

>> b=[1 2 3; 3 2 4; 4 3 5]

b =

     1     2     3
     3     2     4
     4     3     5

>> [b1,c]=matrix1(a,b)

b1 =

     3     4     5

c =

    24     3    18
    12    20    28
    20    45    10

>> |
```

Рисунок 9 – Результат выполнения m-файла matrix1.m

5.4. Варианты заданий.

Задание 1.

1. По двум заданным одномерным массивам формируется третий, содержащий попарно максимальные числа.

2. Дан массив $A(50)$. Определить количество отрицательных элементов, расположенных на четных позициях, и сумму положительных элементов, расположенных на нечетных позициях.

3. Дан массив $A(20)$. Найти минимальный элемент массива, упорядочить элементы, расположенные до этого элемента по убыванию.

4. Дан массив $A(30)$. Если среди элементов массива есть элемент равный некоторому числу M , то переменной X присвоить произведение ненулевых элементов, расположенных до этого элемента, в противном случае переменной X присвоить 0.

5. Дан массив $A(50)$. Найти максимальное количество подряд идущих положительных четных элементов.

6. Дано натуральное число n . Получить последовательность b_1, \dots, b_n , где при $i = 1, 2, \dots, n$ значение b_i равно $2^i + 3^{i+1}$.

7. Дано натуральное число n . Получить последовательность b_1, \dots, b_n , где при $i = 1, 2, \dots, n$ значение b_i равно $1 - \frac{1}{2} + \dots + \frac{(-1)^{i+1}}{i}$.

8. Дано натуральное число n . Получить последовательность b_1, \dots, b_n , где при $i = 1, 2, \dots, n$ значение b_i равно $i \left(\frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{i!} \right)$.

9. Вычислить значения многочлена $x^5 - 9x^4 + 1.7x^2 - 9,6$ для $x = 0, 1, \dots, 5$. Результат выполнения занести в массив чисел.

10. Получить таблицу температур по Цельсию от 0 до 100 градусов и их эквивалентов по шкале Фаренгейта, используя для перевода формулу $t_F = \frac{9}{5}t_C + 32$. Результат выполнения занести в массив чисел t_1, t_2, \dots, t_{100} .

11. Даны действительные числа a_1, a_2, \dots . Известно, что $a_1 > 0$ и что среди a_2, a_3, \dots есть хотя бы одно отрицательное число. Пусть a_1, \dots, a_n – члены данной последовательности, предшествующие первому отрицательному члену (n заранее неизвестно). Получить $a_1 a_2 + a_2 a_3 + \dots + a_{n-1} a_n + a_n a_1$.

12. Даны действительные числа a_1, \dots, a_{37} . Все члены этой последовательности, начиная с первого положительного, уменьшить на 0.5.

13. Даны натуральное число n , действительные числа a_1, \dots, a_n . Получить числа b_1, \dots, b_n , которые связаны с a_1, \dots, a_n следующим образом $b_1 = a_1$, $b_n = a_n$, $b_i = \frac{a_{i+1} - a_i}{3}$, $i = 2, \dots, n-1$.

14. Даны действительные числа a_1, a_2, \dots . Известно, что $a_1 > 0$ и что среди a_2, a_3, \dots есть хотя бы одно отрицательное число. Пусть a_1, \dots, a_n – члены данной последовательности, предшествующие первому отрицательному члену (n заранее неизвестно). Получить $a_1 a_2 \dots a_n$.

15. Дан массив $A(15)$. Найти минимальный элемент среди элементов, расположенных на нечетных позициях массива, а также определить количество и произведение ненулевых элементов, следующих за первым минимальным элементом.

16. Даны целые числа a_1, \dots, a_{30} . Пусть M – наибольшее, а m – наименьшее из a_1, \dots, a_{30} . Получить в порядке возрастания все целые из интервала (m, M) , которые не входят в последовательность a_1, \dots, a_{30} .

17. Дан массив $A(30)$. Найти сумму и количество четных элементов, расположенных между минимальным и максимальным элементами массива.

18. Если у массива $A(20)$ есть элемент, равный квадрату последнего элемента, то все элементы, следующие за ним, возвести в квадрат, иначе вывести массив без изменения.

19. Переменной X присвоить 1, если элементы массива $B(40)$ образуют возрастающую последовательность и значение 0 в противном случае.

20. Дан массив $A(10)$. Найти максимальный элемент массива, упорядочить элементы, следующие за ним по убыванию.

21. Даны действительные числа $a_1, \dots, a_n, b_1, \dots, b_n$. Вычислить $(a_1 + b_n)(a_2 + b_{n-1}) \dots (a_n + b_1)$.

22. Даны действительные числа $a_1, \dots, a_{28}, b_1, \dots, b_{28}$. Члены последовательности c_1, \dots, c_{29} связаны с членами данной последовательностей соотношениями $c_{29}=0$, $c_{29-i} = \frac{a_{29-i}}{b_{29-i} - c_{29-i+1}}$ ($i = 1, \dots, 28$).

23. Даны действительные числа a_1, \dots, a_{20} . Преобразовать эту последовательность по правилу. Большее из чисел a_i и a_{10+i} ($i = 1, \dots, 10$) принять в качестве нового значения a_i , а меньшее – в качестве нового элемента a_{10+i} .

24. Даны действительные числа a_1, \dots, a_n . Получить b_1, \dots, b_{10} , где b_i равно сумме тех членов последовательности a_1, \dots, a_n , которые принадлежат полуинтервалу $(i-1, i]$ ($i = 1, \dots, 10$). Если в полуинтервал не содержит членов последовательности, то соответствующее b_i положить равным нулю.

25. Если у массива $A(30)$ есть элемент, равный B , то переменной X присвоить значение, равное сумме всех положительных четных элементов, предшествующих первому по порядку такому элементу, иначе переменной X присвоить 0.

26. Даны натуральное число n , действительные числа a_1, \dots, a_n . Вычислить
$$-\frac{a_1}{1!} + \frac{a_2}{2!} - \dots + \frac{(-1)^n a_n}{n!}.$$

27. Даны натуральное число n , действительные числа a_1, \dots, a_n . Вычислить $a_1^2, a_1 a_2, a_1 a_3, \dots, a_1 a_n$.

28. Даны натуральное число n , действительные числа a_1, \dots, a_n . Вычислить
$$\left(\sqrt{|a_1|} - a_1\right)^2 + \dots + \left(\sqrt{|a_n|} - a_n\right)^2.$$

29. Даны натуральное число n , действительные числа a_1, \dots, a_n . Вычислить $|a_1|, |a_1 + a_2|, \dots, |a_1 + \dots + a_n|.$

30. Даны натуральное число n , действительные числа a_1, \dots, a_n . Вычислить $a_1, -a_1 a_2, a_1 a_2 a_3, \dots, (-1)^{n+1} a_1 a_2 \dots a_n.$

Задание 2.

1. Дан массив $A[6,6]$. Построить массив $B(6)$ по следующему правилу: $B(1)$ присвоить количество нулевых элементов главной диагонали, $B(2)$ присвоить количество нулевых элементов диагонали, расположенной выше и параллельно главной диагонали и т.д.

2. Дан массив $A[5,5]$. Найти минимальную сумму положительных элементов диагоналей, параллельных побочной диагонали.

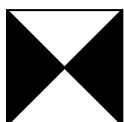
3. Дан массив $A[8,8]$. Найти максимальное произведение ненулевых элементов диагоналей, параллельных главной диагонали.

4. Дана действительная матрица размера $m \times n$. Найти сумму наибольших значений элементов её строк.

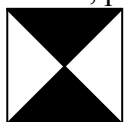
5. Даны натуральное число n , действительная квадратная матрица порядка n , действительные a_1, \dots, a_{n+5} . Элементы последовательности a_1, \dots, a_{n+5} домножить на 10, если наибольший элемент матрицы (в предположении, что такой элемент единственный) находится на главной диагонали, и на 0.5 в противном случае.

6. Дана действительная квадратная матрица порядка n , все элементы которой различны. Найти наибольший элемент среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей.

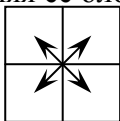
7. Дана действительная квадратная матрица порядка n . Найти наибольшее из значений элементов, расположенных в закрашенной части матрицы:



8. Дана действительная квадратная матрица порядка n . Найти наименьшее из значений элементов, расположенных в закрашенной части матрицы:

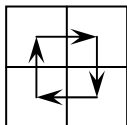


9. Дана действительная квадратная матрица порядка $2n$. Получить новую матрицу, переставляя её блоки размера $n \times n$.



10. Даны квадратные матрицы A и B порядка n . Получить матрицу $AB-BA$.

11. Дана действительная квадратная матрица порядка $2n$. Получить новую матрицу, переставляя её блоки размера $n \times n$.



12. Даны квадратные матрицы A и B порядка n . Получить матрицу $A(B-E)+C$, где E – единичная матрица порядка n , а элементы матрицы C вычисляются по формуле:

$$C_{ij} = \frac{1}{i+j}; (i, j = 1, 2, \dots, n).$$

13. Дана квадратная матрица порядка n . Получить вектор Ab , где b – вектор, элементы которого вычисляются по формуле $(i = 1, \dots, n)$ $b_i = \begin{cases} \frac{1}{i^2+2}, & \text{если } i \text{ четное,} \\ \frac{1}{i}, & \text{нечетное.} \end{cases}$

14. Дана квадратная матрица A порядка n . Получить матрицу AB $(i, j = 1, \dots, n)$; элементы матрицы B вычисляются по формуле $b_{ij} = \begin{cases} \frac{1}{i+j-1}, & \text{если } i < j, \\ 0, & \text{если } i = j, \\ -\frac{1}{i+j-1}, & \text{если } i > j. \end{cases}$

15. Дана действительная матрица порядка $m \times n$. Получить вектор b_1, \dots, b_m , элементы которого соответственно равны значениям средних арифметических элементов строк.

16. Дана действительная матрица порядка $m \times n$. Получить вектор b_1, \dots, b_n , элементы которого соответственно равны разностям наибольших и наименьших значений элементов столбцов.

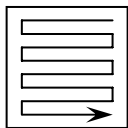
17. Получить квадратную матрицу порядка n :

$$\begin{bmatrix} n & & & & \\ n-1 & n & & & 0 \\ n-2 & n-1 & n & & \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 2 & 3 & \dots & n \end{bmatrix}.$$

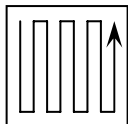
18. Дана действительная квадратная матрица порядка 9. Получить целочисленную квадратную матрицу того же порядка, в которой элемент равен единице, если соответствующий ему элемент исходной матрицы больше элемента, расположенного в его строке на главной диагонали, и равен нулю в противном случае.

19. Даны натуральное число n , целые числа a_1, \dots, a_n и целочисленная квадратная матрица порядка n . Строку с номером i матрицы назовем отмеченной, если $a_i > 0$, и неотмеченной в противном случае. Подсчитать число отрицательных элементов матрицы, расположенных в отмеченных строках.

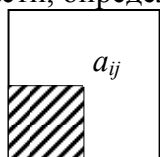
20. Даны действительные числа b_1, \dots, b_{64} . Получить действительную квадратную матрицу порядка 8, элементами которой являются числа b_1, \dots, b_{64} , расположенные в ней по схеме, приведенной на рисунке.



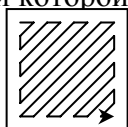
21. Даны действительные числа b_1, \dots, b_{64} . Получить действительную квадратную матрицу порядка 8, элементами которой являются числа b_1, \dots, b_{64} , расположенные в ней по схеме, приведенной на рисунке.



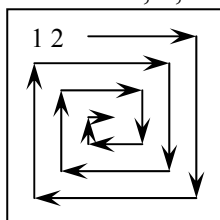
22. Дана действительная квадратная матрица A порядка n . Получить действительную матрицу B порядка n , элемент b_{ij} которой равен произведению элементов данной матрицы, расположенных в области, определяемой индексами i, j так, как показано на рисунке (область заштрихована).



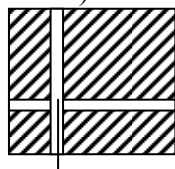
23. Даны действительные числа b_1, \dots, b_{64} . Получить действительную квадратную матрицу порядка 8, элементами которой являются числа b_1, \dots, b_{64} , расположенные в ней по схеме, приведенной на рисунке.



24. Получить целочисленную квадратную матрицу порядка 7, элементами которой являются числа 1, 2, ..., 49, расположенные в ней по спирали.



25. Дана действительная квадратная матрица A порядка n . Получить действительную матрицу B порядка n , элемент b_{ij} которой равен сумме элементов данной матрицы, расположенных в области, определяемой индексами i, j так, как показано на рисунке (область заштрихована).



a_{ij}

26. Дан массив $A[6,6]$. Если среди элементов массива есть элемент, равный B , то переменной X присвоить значение, равное сумме положительных элементов, расположенных слева от этого элемента, иначе переменной X присвоить 0.

27. Дан массив $A[5,5]$. Построить массив $B(5)$ по следующему правилу: $B(J)$ присвоить максимальный элемент J -го столбца массива A .

28. Дан массив $A[7,7]$. Найти произведение и количество четных положительных элементов, расположенных выше главной диагонали.

29. Дан массив $A[6,6]$. Найти суммы положительных элементов строк и присвоить их элементам побочной диагонали соответствующих строк.

30. Дан массив $A[5,5]$. Построить массив $B(5)$ по следующему правилу: $B(I)$ присвоить 1, если в I -й строке массива есть хотя бы один отрицательный элемент, в противном случае $B(I)$ присвоить 0.

5.5. Примерный перечень вопросов на защите работы

1. Каким образом можно вызвать окно редактора/отладчика m-файлов?
2. Перечислите основные кнопки панели редактора/отладчика m-файлов?
3. Какие два типа m-файлов предусмотрено в пакете MATLAB?
4. Перечислите основные свойства m-файлов-сценариев.
5. Какими отличительными свойствами обладает m-файл-функция?
6. Приведите структуру файла-сценария.
7. Приведите структуру файла-функции с одним выходным параметром.
8. Приведите структуру файла-функции с несколькими выходными параметрами.
9. Для чего предназначена команда global?
10. На примере покажите использование подфункций в Matlab.