

## MATLAB-ის გაშვება

პროგრამა MATLAB-ის გაშვება ხდება ისევე როგორც ნებისმიერი პროგრამის გაშვება: ჭინდოფს-ის ოპერაციულ სისტემაში მთავარი მენიუდან (შტარტი), სადაც პროგრამა აღნიშნულია სახელით MATLAB 7 ან Student MATLAB. Linux ან Unix ოპერაციულ სისტემებში საზოგადოდ საკმარისია ტერმინალის ფანჯარაში შევიტანოთ სიტყვა `მატლაბ-ი`. შესაძლებელია აგრეთვე ისარგებლოთ პროგრამის იარლიყით სამუშაო მაგიდაზე, რომლის საშუალებითაც შესრულდება იგივე ამოცანა.

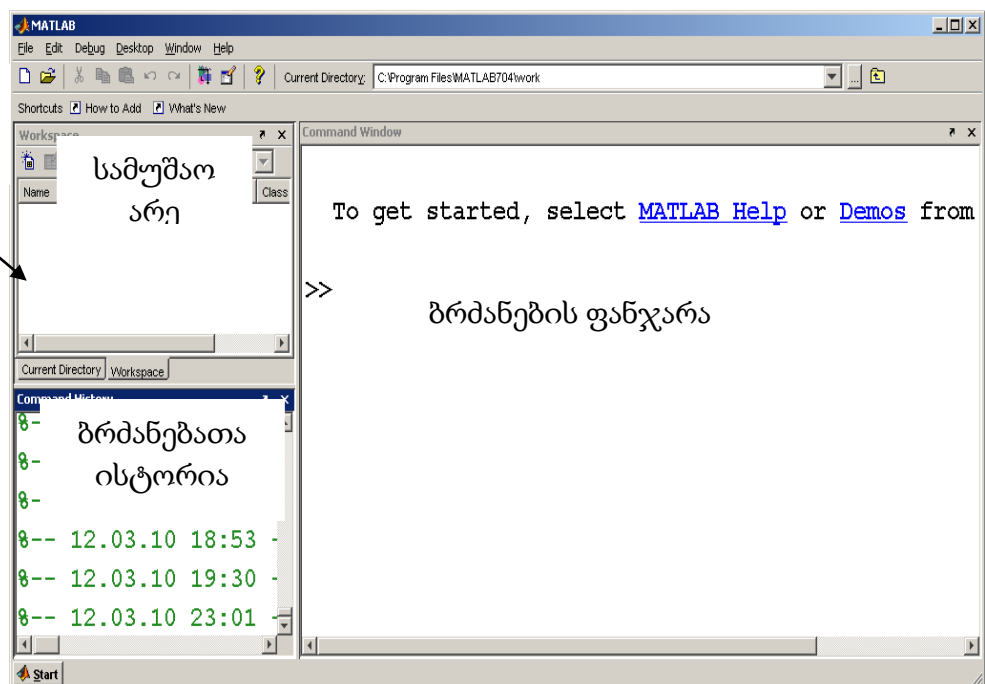
## MATLAB-ის ფანჯრები

პროგრამის გაშვების შედეგად გაიშვება პროგრამა MATLAB-ის ძირითადი ფანჯარა, რომელსაც **სამუშაო მაგიდას** ვუწოდებთ. ეს ფანჯარა შეიცავს სათაურის სტრიქონს (პროგრამის სახელწოდებას - MATLAB), მენიუს სტრიქონს, ინსტრუმენტთა პანელს და ოთხ შიდა ფანჯარას რომელთაგან ერთი დაფარულია. ყველაზე მნიშვნელოვანი ფანჯარაა **Command Window** - ბრძანების ფანჯარა - მარჯვენა მხარეს. დარჩენილი სამი ფანჯარაა: **Command History** - ბრძანებათა ისტორია, **Current Directory** - მიმდინარე საქალაქე, **workspace** - სამუშაო არე.

სამუშაო მაგიდა



მიმდინარე  
საქალაქე

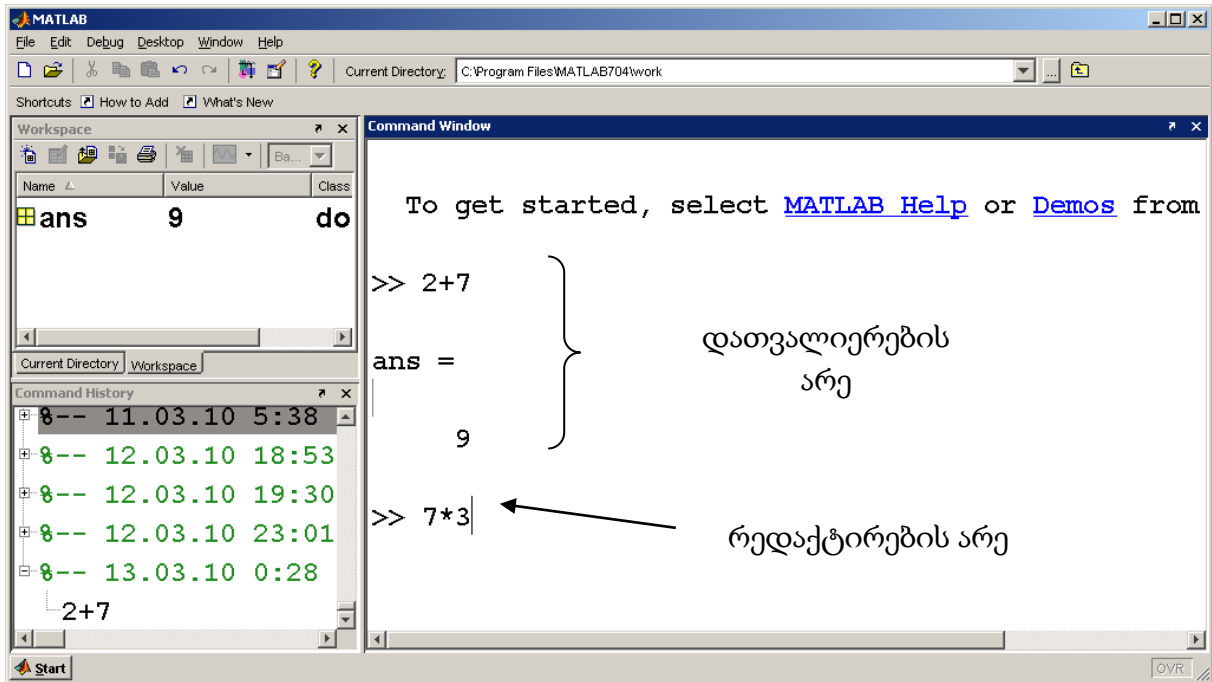


## ბრძანების ფანჯარა (command window)

დასაწყისში გავცნოთ ბრძანების ფანჯარას, რომელიც საშუალებას გვაძლევს შევიყვანოთ ბრძანებები და მივიღოთ შედეგები იქვე ეკრანზე.

ბრძანების ფანჯარა იყოფა ორ არედ: დათვალიერებისა და რედაქტირების არეებად. >> ნიშნის მერე თქვენ შეგიძლიათ შეიტანოთ ახალი ბრძანება ან დაარედაქტიროთ ადრე შესრულებული რომელიმე ბრძანება და ENTER კლავიშის დაჭერით შეასრულოთ ის. უკვე შესრულებული ბრძანებები ხვდებიან დათვალიერების ზონაში, რომლის ჩასწორება შეუძლებელია, მისი ნებისმიერი ფრაგმენტი შეიძლება დავაკოპიროთ ბუფერში და მერე ჩავსვათ სხვა დოკუმენტში.

ბრძანების ფანჯარა სუფთავდება ბრძანებით - **clc**.



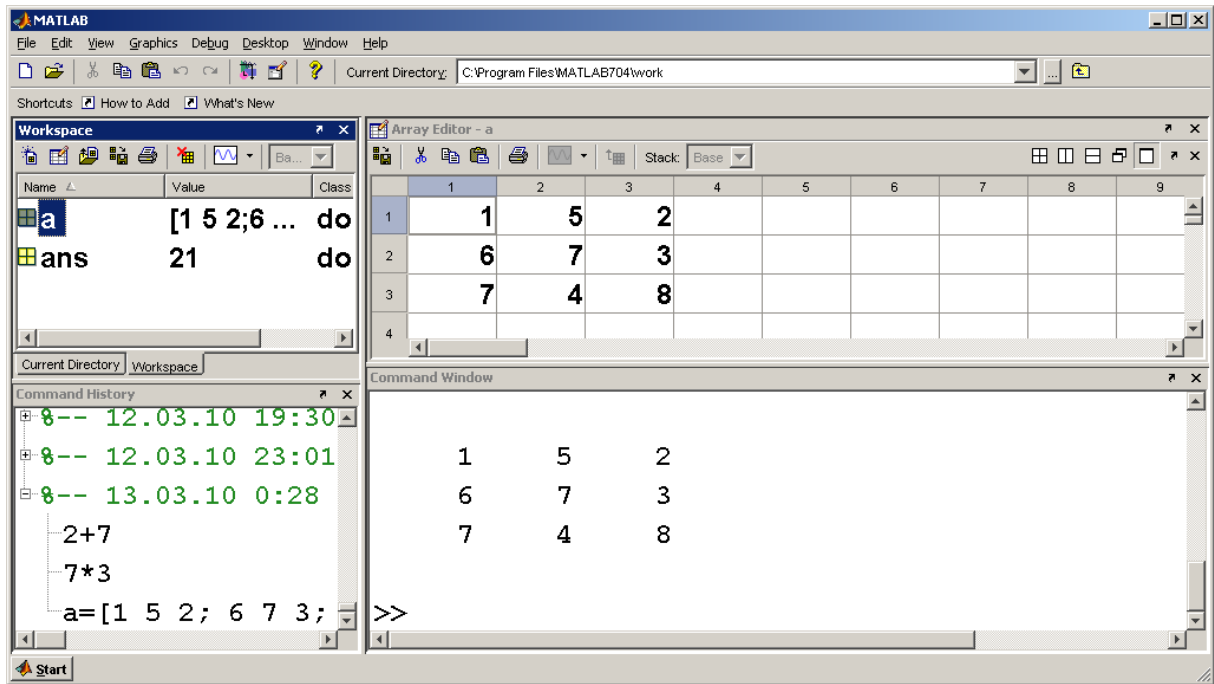
ბრძანების შეტანის სტრიქონის სიგრძე არ აღემატება 256 სიმბოლოს. მისი გაგრძელება რამდენიმე ფიზიკურ სტრიქონად ხდება სამი წერტილის დაბოლოებით ყოველი სტრიქონის ბოლოს და შემდეგ ENTER კლავიშის დაჭერით.

MATLAB-ში შესრულებული ბრძანებების ტექსტური და რიცხვითი მნიშვნელობები ჩნდება იმავე ფანჯარაში, გრაფიკული შედეგი კი აღიბეჭდება სხვა ფანჯარაში.

### ფანჯარა workspace (სამუშაო არე)

ცვლადების მნიშვნელობები, რომლებიც გამოითვლება სამუშაო სეანსის დროს ინახება სპეციალურად გამოყოფილ კომპიუტერის ოპერატიული მეხსიერების არეში, ეგრეთ წოდებულ MATLAB-ის სამუშაო არეში **workspace**.

იმისათვის რომ დავადგინოთ ნებისმიერი ცვლადის მნიშვნელობა საკმარისია ბრძანების სტრიქონში ავკრიფოთ ცვლადის სახელი და დავაჭიროთ **ENTER**-ს, ანდა ფანჯარა workspace-ში, რომელშიც ასახულია მუშაობისას გამოყენებული ყველა ცვლადი, მოვნიშნოთ სასურველი ცვლადი და ორჯერ დავაწკაპუნოთ მასზე თავის მარცხენა ღილაკით. ამის შედეგად ჩნდება ფანჯარა **Array Editor** (მასივის რედაქტორი), რომელშიც შეიძლება დავათვალიეროთ ცვლადები და შევცვალოთ მათი მნიშვნელობები.



### ფანჯარა ჩომანდ HISTORY (ბრძანებათა ისტორია)

ფანჯარაში Command History არის ყველა ის ინფორმაცია რაც შევიტანეთ ბრძანების ფანჯარაში, ეს საშუალებას გვაძლევს დავათვალიეროთ ადრე შესრულებული ყველა ბრძანება. თუ რომელიმე მათგანზე დავაწკაპუნებთ, ის დაუყოვნებლივ შესრულდება ბრძანების ფანჯარაში.

### სამუშაოს დასრულება

სამუშაოს დასრულება, ანუ MATLAB-ის დახურვა, ხდება **quit** ან **exit** ბრძანებით.

### არითმეტიკული ოპერაციების შესრულება

MATLAB შეიძლება გამოვიყენოთ როგორც კალკულატორი. გამოთვლების წარმოების ძირითადი ოპერატორებია:

მინიჭების ოპერატორი =  
 მიმატების ოპერატორი +  
 გამოკლების ოპერატორი -  
 გამრავლების ოპერატორი \*  
 გაყოფის ოპერატორი /  
 ახარისხების ოპერატორი ^ ^^

მაგალითი:

$$9-(5+7)/2+6*1$$

ans =

9

პროგრამა MATLAB-ს გამოაქვს პასუხი და აღნიშნავს მნიშვნელობას ცვლადით ans. შემდგომი გამოთვლების ჩატარებისას შეგიძლიათ გამოიყენოთ ცვლადი **ans**, ოღონდ მისი მნიშვნელობა შეიცვლება ახალი ბრძანების შესრულების შემდეგ, მაგალითისთვის შეგიძლიათ გამოთვალოთ მიღებული პასუხის კვადრატისა და ფესვის ჯამი:

```
ans^2+sqrt(ans)
ans =
    84
```

## მატრიცა, ვექტორი, სკალარი

MATLAB ნებისმიერ ცვლადს აღიქვამს მატრიცის სახით. მატრიცა ეს არის მონაცემთა მწკრივი დალაგებული სტრიქონების და სვეტების სახით. ერთი რიცხვი შეგვიძლია განვიხილოთ როგორც მატრიცა, რომელიც შედგება ერთი სტრიქონისა და ერთი სვეტისაგან. სიბრტყეზე მოცემული წერტილის კოორდინატთა  $x, y$  წყვილი წარმოადგენს მატრიცას ერთი სტრიქონითა და ორი სვეტით, ხოლო  $x-y-z$  კოორდინატების 4 წყება – მატრიცა ოთხი სტრიქონისა და სამი სვეტისაგან. მაგალითად:

$$A = [3.5]$$

$$B = [1.5 \ 3.2]$$

$$C = \begin{bmatrix} -1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

თუ მატრიცა შეიცავს ერთ სტრიქონსა და ერთ სვეტს, იგი სკალარია. თუ იგი შეიცავს 1 სტრიქონს და რამდენიმე სვეტს – სტრიქონ ვექტორია, ხოლო თუ შეიცავს რამდენიმე სტრიქონსა და 1 სვეტს – იგი სვეტ ვექტორია.

მატრიცის ელემენტებს მიუთითებენ ორი ინდექსით – სტრიქონისა და სვეტის ნომერით. მაგ. C მატრიცის მეოთხე სტრიქონის და მესამე სვეტის ელემენტი C(4,3) არის 2. სტრიქონ-ვექტორისა და სვეტ-ვექტორის შემთხვევაში დასაშვებია მხოლოდ ერთი ინდექსის მითითებაც.

მატრიცის ზომა განისაზღვრება სტრიქონების და სვეტების რაოდენობით. მაგალითად C მატრიცის ზომაა  $4 \times 3$ . თუ მატრიცის სვეტების და სტრიქონების რაოდენობა ერთიდაიგივეა, მას კვადრატულს უწოდებენ.

ბრძანება **size** საჭიროა თუ გვსურს გავიგოთ რომელიმე ცვლადის ზომა. მაგალითად სიზე(8) მოგვცემს შემდეგ შედეგს

```
ans=
    4    3
```

## სავარჯიშო

უპასუხეთ კითხვებს შემდეგი მატრიცის მაგალითზე:

$$C = \begin{bmatrix} 0.6 & 1.5 & 2.3 & -0.5 \\ 8.2 & 0.5 & -0.1 & -0.2 \\ 5.7 & 8.2 & 9.0 & 1.5 \\ 0.5 & 0.5 & 2.4 & 0.5 \\ 1.2 & -2.3 & -4.5 & 0.5 \end{bmatrix}$$

1. რა ზომისაა მატრიცა?
2. არის თუ არა იგი კვადრატული?
3. დაწერეთ აღნიშვნა მატრიცის ელემენტებისა, რომელთა მნიშვნელობა 0.5 ტოლია

#### 4. დაწერეთ აღნიშვნა მატრიცის ელემენტებისა, რომელთა მნიშვნელობა უარყოფითია

### მატრიცის ინიციალიზაცია

როგორ განვსაზღვროთ მატრიცა MATLAB-ში? განვიხილავთ ოთხ სხვადასხვა მეთოდს.

პირველი მეთოდია მატრიცის ელემენტების ცხადად განსაზღვრა, მეორე – მატრიცის ფაილიდან ამოკითხვა, მესამე – ორწერტილის (:) გამოყენება, მეოთხე – მონაცემთა შეყვანა კლავიატურიდან.

**პირველი მეთოდი – მატრიცის ცხადად განსაზღვრა.** ამისათვის მატრიცას ელემენტებს ათავსებენ კვადრატულ ფრჩხილებში:

$A = [3.5];$

$B = [1.5, 3.1];$

$R = [-1, 0, 0; 1, 1, 0; 1, -1, 0; 0, 0, 2];$

ეს ბრძანებები მაგალითია იმისა, თუ როგორ შევქმნათ სხვადასხვა ზომის მატრიცა.

მატრიცის, ანუ ცვლადის, სახელი უნდა იწყებოდეს ასოითი გამოსახულებით და შედგებოდეს არაუმეტეს 19 ნიშნისაგან (დასაშვებია მხოლოდ ციფრების, ასოებისა და ქვედა ტირეს გამოყენება) და უნდა ეწეროს მინიჭების ოპერატორის მარცხნივ. ტოლობის ნიშნის მარჯვენა მხარე წარმოადგენს კვადრატულ ფრჩხილებში ჩასმულ მატრიცის ელემენტების მნიშვნელობებს, წერტილმძიმით სტრიქონები გამოიყოფა ერთმანეთისაგან, ხოლო სტრიქონებში მონაცემები შეიძლება გამოვყოთ მძიმით. გამოსახულება შეიძლება შეიცავდეს + ან – ნიშანს.

ათწილადის ნიშანად გამოიყენება “.” და არა “/”.

როდესაც განვსაზღვრავთ მატრიცას, თუ ბოლოში არ დავუსვამთ წერტილ-მძიმეს, MATLAB ბრძანების ფანჯარაში გამოგვიყვანს მის გამოსახულებას. სცადეთ შეიყვანოთ ზემოთაღნიშნული მონაცემები ბოლოში წერტილ-მძიმის დაურთველად.

მატრიცა შეგვიძლია განვსაზღვროთ ისე, რომ სტრიქონები წერტილ-მძიმით კი არ გამოვყოთ ერთმანეთისაგან, არამედ გადავიტანოთ შემდეგ ხაზზე “ENTER” კლავიშის საშუალებით.

$C = [-1\ 0\ 0$   
 $\quad 1\ 1\ 0$   
 $\quad 1\ -1\ 0$   
 $\quad 0\ 0\ 2];$

თუ მატრიცის სტრიქონში ბევრი ელემენტებია, შეგვიძლია გარკვეული რაოდენობით ელემენტების შემდეგ დაწეროთ სამი წერტილი და სტრიქონი გავაგრძელოთ შემდეგ ხაზზე. მაგალითად:

$F = [1\ 0\ 5\ 7\ 3\ 9\ 6\ 7\ 8\ 0\ 4];$

იგივეა რაც

$F = [1\ 0\ 5\ 7\ 3\ \dots$   
 $\quad 9\ 6\ 7\ 8\ 0\ 4];$

MATLAB საშუალებას იძლევა განვსაზღვროთ ახალი მატრიცა უკვე შექმნილის ბაზაზე. მაგალითად განვიხილოთ შემდეგი ბრძანებები:

$$B = [1.5, 3.1];$$

$$S = [3.0 \ B];$$

ეს იგივეა, რაც:

$$S = [3.0 \ 1.5 \ 3.1];$$

ასევე შეგვიძლია შევცვალოთ მატრიცის ელემენტების მნიშვნელობები, ან დავუმატოთ ახალი ელემენტები:

$$S(2) = -1.0;$$

ეს ბრძანება შეცვლის 2-ე მატრიცის მეორე ელემენტს 1.5-ს, მიაჩნებს რა მას ახალ მნიშვნელობას -1.0-ს.

ასევე შესაძლებელია მატრიცას დავუმატოთ ახალი ელემენტები:

$$S(4) = 5.5;$$

ამის შედეგად სამედიანტიანი მატრიცა ოთხედიანად იქცევა. მაგრამ თუ მივცემთ ბრძანებას:

$$S(8) = 9.5;$$

მატრიცა გახდება 8 ელემენტიანი, მერვე ელემენტს მიენიჭება 9.5, მეხუთე, მეექვსე და მეშვიდე ელემენტები კი მიიღებენ 0-ის ტოლ მნიშვნელობებს.

### სავარჯიშო

განსაზღვრეთ მატრიცის ზომა და შეამოწმეთ იგი MATLAB-ის საშუალებით:

1.  $A = [1, 0, 0, 0, 0, 1];$

2.  $B = [2; 4; 6; 10];$

3.  $C = [5 \ 3 \ 6; 6 \ 2 \ -3];$

4.  $E = [3 \ 5 \ 10 \ 0; 0 \ 0 \ \dots \\ 0 \ 3; 3 \ 9 \ 9 \ 8];$

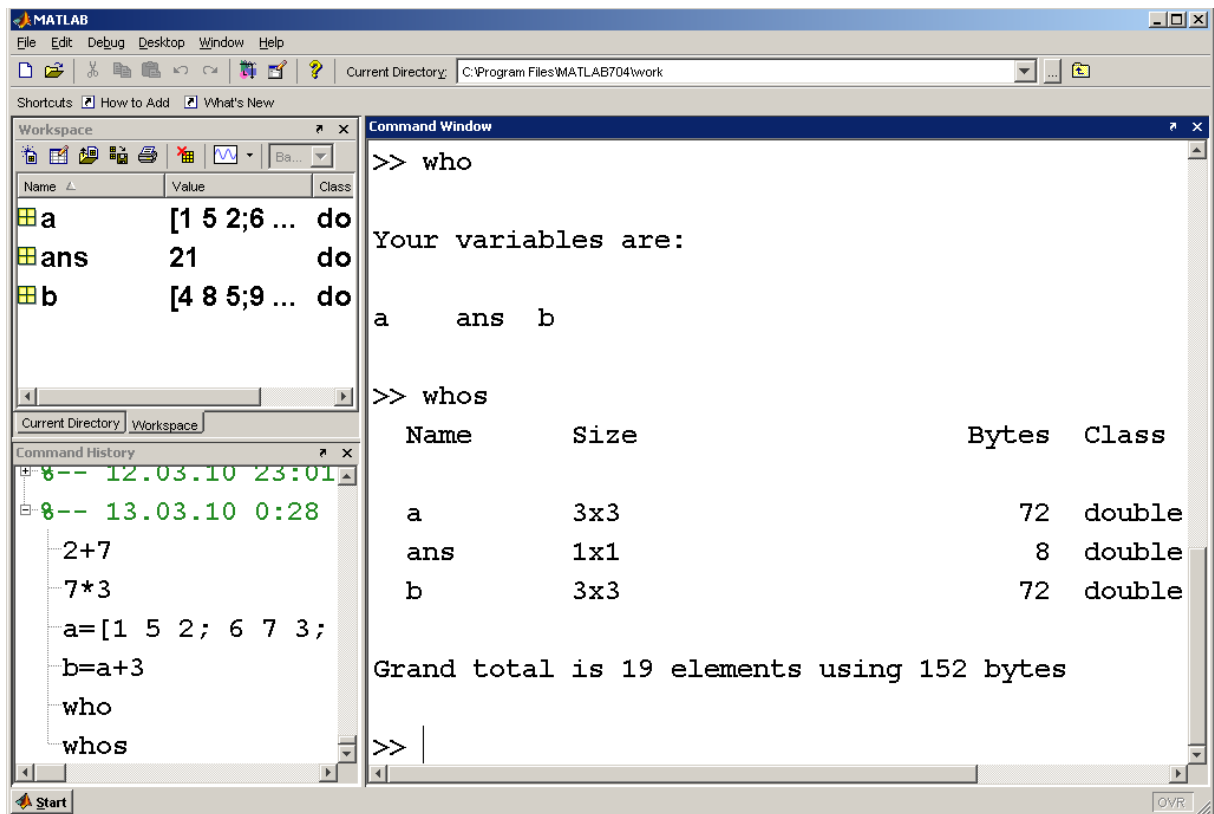
5.  $T = [4 \ 24 \ 9];$

$$Q = [\text{თ} \ 0 \ \text{თ}];$$

6.  $V = [\text{ჩ}(2,1); B];$

### მეორე მეთოდი – მატრიცის ფაილიდან ამოკითხვა.

იმისათვის, რომ გავიგოთ ჩვენს მიერ შექმნილი ცვლადების და მატრიცების სტატუსი MATLAB-ს აქვს რამდენიმე ბრძანება. ბრძანება **who** გვამღებს ცვლადების სიას, რომელიც შევქმენით, ხოლო **whos** დამატებით ინფორმაციასაც იძლევა მათი ზომის, მანქანური მოცულობის და კლასის შესახებ.



მატრიცა შეგვიძლია შევქმნათ აგრეთვე იმ ინფორმაციის საშუალებით, რომელიც ჩაწერილია ნებისმიერ ტექსტურ ფაილში ან .mat გაფართოების ფაილებიდან, რომელიც შეიცავს ინფორმაციას ორობით ფორმატში.

.მატ გაფართოების ფაილები იქმნება MATLAB-ის მიერ **save** ბრძანების საშუალებით. მაგალითად ბრძანება

```
save data1 x y;
```

შეინახავს x და y მონაცემებს ფაილში data1.mat. .mat გაფართოებას პროგრამა ავტომატურად დაურთავს ფაილის სახელს.

ამ მონაცემთა სამუშაო სივრცეში გამოსაძახებლად ვსარგებლობთ ბრძანებით:

**Load data1;**

**მესამე მეთოდი - ორწერტილის (:) გამოყენება**

ახალი მატრიცის საწარმოებლად შეგვიძლია გამოვიყენოთ : - ორწერტილი. თუ იგი დგას ორ მთელ რიცხვს შორის, შეიქმნება ვექტორი, რომლის ელემენტები იქნება ყველა მთელი რიცხვი, რომელიც ამ ორ რიცხვს შორის მდებარეობს, მათი ჩათვლით. მაგალითად:

```
H = 1:8
```

იძლევა ვექტორს

```
H = [1 2 3 4 5 6 7 8];
```

თუ ორწერტილოვანი ოპერატორი 3 რიცხვს გამოყოფს ერთმანეთისაგან, მაშინ შეიქმნება ვექტორი, რომლის ელემენტები იქნება რიცხვები პირველსა და მესამე რიცხვს შორის, ნაზრდით მეორე რიცხვი:

```
time = 0.0:0.5:5.0;
```

გვაძლევს

```
time = [0.0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5];
```

MATLAB –ში ზოგჯერ გვჭირდება ცარიელი მატრიცა, იგი განსხვავდება მატრიცისაგან, რომლის ყველა ელემენტი ნულის ტოლია. მაგალითად:

```
a = [];
```

```
b = 4 : -1 : 5;
```

ორივე ბრძანება ქმნის ცარიელ მატრიცას.

გამოსახულება C(:) თუ C მატრიცაა, ქმნის ერთ გრძელ სვეტ ვექტორს, როლის ელემენტებია C მატრიცის პიველი სვეტის ელემენტები, რომელსაც მოყვება მეორე სვეტის ელემენტები და ა. შ.

### მეთოდე მეთოდი \_ მონაცემთა შეყვანა კლავიატურიდან

შესაძლებელია მატრიცის მნიშვნელობები შევიყვანოთ პირდაპირ კლავიატურიდან **input** ბრძანების საშუალებით, რომელსაც ეკრანზე გამოჰყავს ტექსტი და ელოდება მონაცემებს. მომხმარებლის მიერ მიწოდებული მონაცემებით შეიქმნება შესაბამისი მატრიცა. თუ არ შევიყვანთ არანაირ მონაცემს, შეიქმნება ცარიელი მატრიცა. თუ ბრძანება არ მთავრდება წერტილ-მძიმით, შეყვანილი მონაცემების საფუძველზე შექმნილი მატრიცა დაიბეჭდება ბრძანებათა ფანჯარაში.

განვიხილოთ ბრძანება:

```
z = input('Enter value for z');
```

ეს ბრძანება მოგვცემს ეკრანზე ტექსტს: Enter value for z. უნდა შევიყვანოთ მონაცემები [5.1 6.3 -18.0], რომელიც მიენიჭება z. რადგან ბრძანება მთავრდება წერტილმძიმით, შეყვანილი z მნიშვნელობა არ გამოჩნდება ეკრანზე. თუ მითითებული ტექსტის ბოლოს იქნება ნიშანი \n, ბრძანების ფანჯარაში ტექსტის გამოყვანის შემდეგ მოხდება ახალ სტრიქონზე გადასვლა და ჩვენს მიერ შეყვანილი რიცხვები ეკრანზე გამოყვანილი ტექსტის ქვევით და არა გასწვრივ როგორც იყო წინა ბრძანების შესრულებისას.

### სავარჯიშო

იპოვეთ ზომა და შემადგენლობა შემდეგი მატრიცებისათვის, თუ :

$$G = \begin{bmatrix} 0.6 & 1.5 & 2.3 & -0.5 \\ 8.2 & 0.5 & -0.1 & -2.0 \\ 5.7 & 8.2 & 9.0 & 1.5 \\ 0.5 & 0.5 & 2.4 & 0.5 \\ 1.2 & -2.3 & -4.5 & 0.5 \end{bmatrix}$$

1. A = G(:, 2);
2. B = G(4, :);
3. C = [10 : 15];
4. D = [4 : 9; 1 : 6];
5. E = [-5, 5];
6. F = [0.0 : 0.1 : 1.0];
7. T1 = G(4 : 5, 1 : 3);
8. T2 = G(1 : 2 : 5, :);

### გრაფიკების აგება

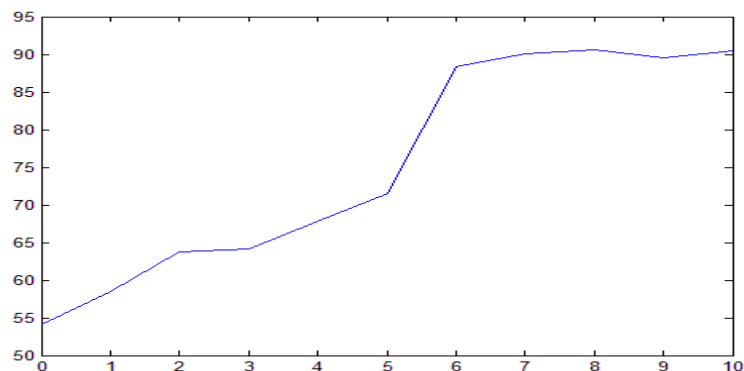


შემდგომში დაწვრილებით განვიხილავთ MATLAB-ის გრაფიკულ შესაძლებლობებს. ახლა კი იმის საილუსტრაციოდ თუ რაოდენ მარტივი და მოხერხებულია გრაფიკის აგება MATLAB-ში ავავთ გრაფიკი x და y ვექტორების სახით წარმოდგენილი მონაცემების მიხედვით.

დავუშვათ გვინდა ავავთ ტემპერატურის მონაცემები, რომელიც ექსპერიმენტის ჩატარების დროს შევკრიბეთ.

ასევე დავუშვათ, რომ დრო მოცემული გვაქვს როგორც x ვექტორის, ხოლო ტემპერატურა – y ვექტორის ელემენტები. (ეს ვექტორები წინასწარ უნდა შევქმნათ ან ფაილიდან გამოვიძახოთ ჩვენს სამუშაო სივრცეში). გრაფიკის ასაგებად ვიყენებთ ბრძანებას **plot**, არგუმენტით x, y, სადაც ორივე ან სვეტი ვექტორია, ან სტრიქონი.

დრო, წამებში	ტემპერატურა, ფარენჰეიტის გრადუსებში
0	54.2
1	58.5
2	63.8
3	64.2
4	67.8
5	71.5
6	88.3
7	90.1
8	90.6
9	89.5
10	90.4



`plot(x,y)`

ავტომატურად შეიქმნება გრაფიკი. იმისათვის, რომ გრაფიკი უფრო ინფორმაციული იყოს, გავუკეთოთ მას სათაური, ღერძებს დავაწეროთ შესაბამისი სახელები და დავიტანოთ ნახაზზე საკოორდინატო ბადე ნახ. 2.

`plot(x,y)`

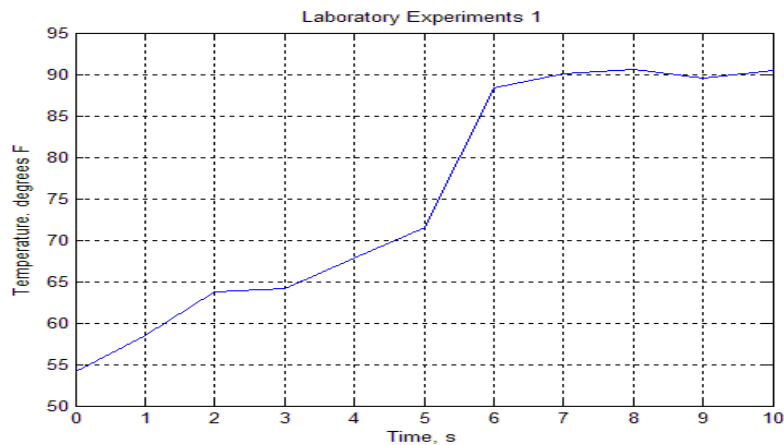
`title('Laboratory Experiments 1')`

`xlabel('time, s')`

```
ylabel('Temperature, degrees F')
grid
```

სამი წერტილი ყოველი სტრიქონის ბოლოს საჭიროა იმისათვის, რომ MATLAB – მა ეს ხუთივე ბრძანება ერთდროულად გაუშვას. ამის მაგივრად შეგვეძლოს თითოეული ბრძანება ცალ-ცალკე მიგვეცა ერთმანეთის მიყოლებით.

თუ ერთიდაიგივე პროგრამაში გვინდა ავაგოთ ორი ნახაზი, ან ავაგოთ გრაფიკი და გავაგრძელოთ შემდგომი გამოთვლები, MATLAB გამოიყენებს ნახაზს და გააგრძელებს პროგრამაში მითითებულ ბრძანებებს, მაგრამ თუ გვინდა, რომ ნახაზს შევხედოთ, ვიდრე პროგრამა გააგრძელებს მუშაობას, **plot** ბრძანების შემდეგ უნდა დავწეროთ ბრძანება **pause**. ასეთ შემთხვევაში MATLAB დაელოდება ჩვენს დასტურს (**ENTER** კლავიშის დაჭერას) პროგრამის გაგრძელებისათვის. გრაფიკული ფანჯრის გასუფთავება ხდება **clf** ბრძანებით. პლოტ ბრძანების შესაძლებლობის უკეთ გასაცნობად ავკრიფოთ **help plot**. **help**-ის გამოყენებით შეგვიძლია მივიღოთ ნებისმიერი მატლაბის ბრძანების ან ფუნქციის გამოყენების ინსტრუქცია.



## m – ფაილები

გარდა იმისა, რომ ბრძანებები შეგვიძლია მივცეთ ბრძანებების ფანჯრიდან, შეგვიძლია წინასწარ შევქმნათ ბრძანებათა მწკრივი ტექსტური ფაილის სახით, რომლის სახელს უნდა ჰქონდეს გაფართოება **.m**, მაგ. ლაბტესტ.მ ასეთ ფაილებს ეწოდება **m-ფაილები**. ფაილის სახით შენახული ბრძანებათა ასეთი მწკრივი შეგვიძლია შემდგომშიც გამოვიყენოთ. ამ ფაილებს აგრეთვე უწოდებენ ხელნაწერ ფაილებს, რადგან წარმოადგენს ტექსტურ ASCII ფაილს და შეგვიძლია შევქმნათ ნებისმიერი ტექსტური რედაქტორით. თუ გვინდა ბრძანებების ფანჯარაში გამოვიყენოთ **.m** ფაილის ტექსტი უნდა ვისარგებლოთ ბრძანებით **type**.

**m** ფაილები ასევე გამოიყენება MATLAB-ის ახალი ფუნქციების შესაქმნელად.

არსებობს ორი ტიპის m-ფაილები: **m-ფაილი-სცენარები** და **m-ფაილები-ფუნქციები**.

**m-ფაილი-სცენარები** შეიცავენ პროგრამა MATLAB-ის ბრძანებათა მიმდევრობას რომლებიც ზუსტად ამ მიმდევრობით, ანუ მოცემული სცენარით, სრულდება. მათ შეგვიძლიათ მიაწოდოთ ნებისმიერი სახელი (სახელი უნდა იწყებოდეს ასოითი გამოსახულებით და დასაშვებია მხოლოდ ციფრების, ასოებისა და ქვედა ტირეს გამოყენება), მხოლოდ გაფართოება აუცილებლად უნდა იყოს **.m**. ამ სცენარის ანუ

პროგრამის გაშვება შეიძლება ბრძანებათა ფანჯარაიდან მისი სახელის აკრეფითა და Enter კლავიშის დაჭერით ან m-ფაილების რედაქტორიდან **run** ბრძანებით ან F5 ღილაკზე დაწკაპუნებით.

რეკომენდირებულია რომ m-ფაილებს დაურთოდ კომენტარები. კომენტარები იწყება პროცენტის ნიშნით %; **MATLAB**-ის რედაქტორში კომენტარები მწვანე ფერი აქვთ რაც ძლიერ მოხერხებულია მათი შავი ფერის პრძანებათა ტექსტიდან გამოსარჩევად. შემდგომში ჩვენ გავეცნობით ამ რედაქტორის სხვა დადებით მხარეებსაც.

m-ფაილ-ფუნქციის პირველი სტრიქონი იწყება სიტყვით **function** (რედაქტორში მოიცემა ლურჯი ფერით), რომლის შემდეგ მოდის კვადრატულ ფრჩხილებში მოთავსებული ფუნქციის მიერ გამოთვლილი სამუშაო არეში დასაბრუნებელი ცვლადების ჩამონათვალი, შემდეგ მინიჭების ოპერატორი = და ფუნქციის სახელი და ბოლოს ფუნქციაში სამუშაო არედან შემავალი მრგვალ ფრჩხილებში მოთავსებული ცვლადთა სია. m-ფაილ-ფუნქციის სახელი უნდა ემთხვეოდეს ფუნქციის სახელს. m-ფაილ-ფუნქციებში შესრულებული მოქმედებები და შემოყვანილი ცვლადები, m-ფაილ-სცენარებიდან განსხვავებით, არ აისახებიან არც ბრძანებათა ფანჯარაში და არც სამუშაო არეში, ანუ ისინი სრულდებიან ავტონომიურად და სამუშაო არედან იღებენ მხოლოდ შემავალ ცვლადებს და აბრუნებენ მხოლოდ გამომავალ ცვლადებს.

ბრძანება **what** გვაძლევს იმ ფაილების სიას, რომლების განთავსებულია მიმდინარე დირექტორიაში. ბრძანება **type** გვაძლევს მითითებული ფაილის შინაარსს.

### სპეცისიმბოლოები:

[ ]	გამოიყენება მატრიცის შესაქმნელად
( )	მატრიცის ელემენტების იდექსებისათვის
,	გამოყოფს ერთმანეთისაგან ინდექსებსა და მატრიცის ელემენტებს
;	გამოყოფს მატრიცებს ან ბრძანებებს, კრძალავს ბეჭდვას
>>	მოგვიხმობს შემდეგი ბრძანებისაკენ
...	აგრძელებს ბრძანებას შემდეგ ხაზზე
%	განსაზღვრავს კომენტარს ან ფორმატს
:	გამოიყენება მატრიცის შესაქმნელად
\n	განსაზღვრავს ახალ ხაზზე გადასვლას
=	მინიჭების ოპერატორი
+	მიმატების ოპერატორი
-	გამოკლების ოპერატორი
*	გამრავლების ოპერატორი
/	გაყოფის ოპერატორი
^	ხარისხში აყვანის ოპერატორი

### ბრძანებები და ფუნქციები

ans	გამოთვლის შედეგის შემცველი ცვლადი, თუ შედეგის მისანიჭებელი ცვლადი არაა მითითებული
clc	ასუფთავებს ბრძანებების ფანჯარას
clear	ასუფთავებს სამუშაო სივრცეს
clf	ასუფთავებს გრაფიკულ ფანჯარას
disp	გამოყავს ეკრანზე მატრიცა ან ტექსტი
exit	ხურავს MATLAB-ს
grid	გრაფიკულ გამოსახულებაზე გამოჰყავს საკოორდინატო ბადე
help	გამოიძახებს MATLAB ჰელპ

input	საშუალებას გვაძლევს მონაცემი შევიყვანოთ კლავიატურიდან
load	გამოიძახებს მატრიცას ფაილიდან
pause	დროებით წყვეტს პროგრამას
plot	აგებს გრაფიკს
quit	ხურავს MATLAB-ს
save	ინახავს ცვლადს .მატ გაფართოების ფაილში
size	განსაზღვრავს მატრიცის ზომას
title	გრაფიკული გამოსახულების სათაური
type	დაგვიბეჭდავს ფაილის შინაარსს
what	ჩამოგვითვლის m ფაილებს მოც. სამუშაო სივრცეში
who	ჩამოთვლის ცვლადებს მოც. სამუშაო სივრცეში
whos	ჩამოთვლის ცვლადებს მოც. სამუშაო სივრცეში და მიუთითებს მათ ზომებს
xlabel	ტექსტის მიწერა გრაფიკის x ღერძზე
ylabel	ტექსტის მიწერა გრაფიკის y ღერძზე