درس

برنامه نویسی کامپیوتر

Computer programming

if . . . (else . . .) end – دستور if برنامه را قادر می سازد که تصمیم بگیرد که چه دستورهایی باید اجرا گردند. مثال:

```
x = input(' x = ');
if x >= 0
   y=x^2
end
```

عبارتی که به دنبال کلمه if می آید باید یک عبارت منطقی باشد. در صورت درست بودن این عبارت منطقی، دستورهایی که در سطرهای بین if و end قرار دارند بترتیب اجرا می گردند و در صورت نادرست بودن این عبارت منطقی، دستورهای گفته شده نادیده گرفته می شوند.

شما همچنین می توانید از دستور else استفاده کنید. مثال:

```
x = input(' x = ');
if x >= 0
   y=x^2
else
   y=-x^2
end
```

۲-۵ آرایه و ماتریس

در نرم‌افزار *MATLAB*، اساساً همه پارامترها و اعداد بصورت یک آرایه و ماتریس شناخته می‌شوند. به همین لحاظ آشنایی با نحوه تشکیل آن دارای اهمیت خاصی است. برای نمایش آرایه و ماتریس از نماد «[]» براکت (کروشه) استفاده می‌گردد. همچنین برای جداکردن عناصر ستون‌ها در یک آرایه و ماتریس از نماد «,» کاما (ویرگول) و یا فاصله (فضای خالی) و برای جداکردن عناصر سطرها از «;» سمیکولون و یا فشردن کلید *Enter* استفاده می‌شود. بعنوان مثال آرایه A در دو حالت سطری و ستونی بصورت زیر است:

```
>> A=[ 1 2 3]
A =
     1     2     3

>> A=[1;2;3]
A =
     1
     2
     3
```

۳-۵-۲ جستجوی عناصر و تغییر آنها

برای مشخص کردن تعداد عناصر یک آرایه و ماتریس می‌توان از دستورات *size* و *length* استفاده کرد. بعنوان مثال فرمان *size(b)* تعداد سطرها و ستون‌های ماتریس b را مشخص می‌کند:

```
>> size(b)
ans =
     3     3
```

ولی اگر بخواهیم عدد مربوط به تعداد سطرها و ستون‌ها را در متغیری ذخیره نماییم، باید از دستور size بصورت ذیل استفاده گردد:

```
>> [m n]=size(a)
m =
     1
n =
     5
```

همچنین برای مشخص‌کردن تنها تعداد سطرهای یک ماتریس از دستور *size(a,1)* و نیز برای یافتن تعداد ستون‌های آن از دستور *size(a,2)* (1 معرف سطر و 2 نشان‌دهنده ستون) استفاده می‌شود.

```
>> size(a,1)
ans =
     1
>> size(a,2)
ans =
     5
```

همچنین برای تغییر عناصر یک آرایه و ماتریس، تنها کافیست با مشخص‌کردن آدرس موردنظر مقادیر آنها را با مقادیر جدید جایگزین کرد. بعنوان مثال دستور ذیل ستون اول ماتریس b را برابر 10 قرار می دهد.

```
>> b(:,1)=10
b =
    10     2     3
    10     5     6
    10     8     9
```

و یا دستور مقابل درایه سطر دوم و ستون سوم را با مقدار 20 جایگزین می‌کند.

```
>> b(2,3)=20
b =
    10     2     3
    10     5    20
    10     8     9
```

همچنین برای حذف یک سطر یا ستون می‌توان از نماد *[]* استفاده کرد. بعنوان مثال با استفاده از دستور ذیل سطر سوم ماتریس b حذف می‌گردد:

```
>> b(3,:)=[]
b =
    10     2     3
    10     5    20
```

۲-۵-۶ محاسبات ماتریسی

برای ماتریس‌ها نیز می‌توان از عملگرهای جمع، ضرب و ... استفاده کرد. فقط باید توجه داشت که باید قوانین مربوط به جبرخطی در عملیات محاسباتی به درستی رعایت گردد. بعنوان مثال اگر دو ماتریس A و B بصورت ذیل مشخص باشند، با استفاده از عملگرهای $A+B$، $A-B$، $A*B$ و A/B می‌توان نتیجه محاسبات را مشاهده نمود.

```
>> A=[2 4 1;3 9 5;6 1 7];
>> B=[3 5 1;6 2 7;3 4 2];

>> A+B
ans =
     5     9     2
     9    11    12
     9     5     9

>> A*B
ans =
    33    22    32
    78    53    76
    45    60    27
```

همچنین برای حل مسائل مختلف جبرخطی می‌توان از توابع مندرج در جدول ذیل نیز استفاده کرد:

عملکرد	تابع	عملکرد	تابع
دترمینان ماتریس	det(A)	رنک یا مرتبه ماتریس	rank(A)
معکوس ماتریس	inv(A)	نُرم ماتریس	norm(A)
تریس ماتریس	trac(A)	نُرم‌های دیگر *m=1,… ,p, … ,inf*	norm(A,m)
ترانهاده ماتریس	A'	ریشه‌دوم ماتریس	Sqrtm(A)

بعنوان مثال برنامه‌ای که مقدار a و b را از ورودی دریافت کرده و معادله زیر را حل نماید، بصورت ذیل است:

$$\begin{aligned} 2x+y &= a \\ x+3y &= a-b \\ 3x-2y &= 2b \end{aligned}$$

برای حل این‌گونه مسائل ابتدا باید آن را به فرم ماتریسی ($AX=L$) تبدیل نماییم:

$$\begin{bmatrix} 2 & 1 \\ 1 & 3 \\ 3 & -2 \end{bmatrix} \times \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a \\ a-b \\ 2b \end{bmatrix} \rightarrow AX = L$$

۲-۸-۱ خواندن از فایل

برای خواندن داده‌ها از فایل موردنظر توابع مختلفی وجود دارد که برای نمونه می‌توان از دو دستور *load* و *textread* استفاده کرد. اگر فایل ورودی فقط شامل اعداد و ارقام باشد، می‌توان تمام داده‌ها را در یک ماتریس با استفاده از فرمان *load* ذخیره کرده و پردازش‌های لازم را انجام داد. بعنوان مثال اگر فایل *Data1.txt* شامل اعداد زیر باشد:

```
Data1.txt
     10.2 15.1 20.8
     30.1 15.6 36.7
     15.7 18.9 20.3
     33.9 18.7 66.1
```

در این صورت با استفاده از دستور ذیل می‌توان تمام داده‌ها را در متغیر ماتریسی A ذخیره کرده و با آدرس‌دهی مناسب به هریک از آنها دسترسی داشت:

```
>> A=load('Data2.txt')
A =
   10.2000   15.1000   20.8000
   30.1000   15.6000   36.7000
   15.7000   18.9000   20.3000
   33.9000   18.7000   66.1000

>> A(2,3)
ans =
   36.7000
```