فصل ۱۰

مثال‌های متنوع از برنامه‌نویسی

مثال ۱.۱۰
در مثال زیر می‌خواهیم که خروجی ماتریس $J(x)$ را با داشتن دو متغیر $x_۱$ و $x_۲$ بدست آوریم.

$$J(x) = \begin{bmatrix} x_۱ + x_۲ & (x_۱ + x_۲)^۲ \\ x_۱ x_۲ & \sqrt{x_۱ + x_۲} \end{bmatrix}, \quad x = \begin{bmatrix} x_۱ \\ x_۲ \end{bmatrix}$$

```
% This program returns a 2x2 matrix
% J(x)=[f1,f2;f3,f4]
%With f1, f2, f3, f4 are functions of x1 and x2
x=input('vector x=[x1;x2] \n x=');
J=zeros(2,2);
J(1,1)=x(1)+x(2);
J(1,2)=(x(1)+x(2))^2;
J(2,1)=x(1)*x(2);
J(2,2)=sqrt(x(1)+x(2));
disp(J)
```

مثال ۲.۱۰
در این مثال می‌خواهیم با استفاده از دستور If مساحت یک دایره را محاسبه کنیم. $S = \pi r^۲$

```
% This script calculates the area of a circle.
radius=input('Please enter the radius: \n');
if radius <= 0
    disp('Sorry, This is not a valid radius')
else
    area=pi*radius^2;
    disp(['The area is: ', num2str(area)])
end
```

مثال ۳.۱۰
در این مثال می‌خواهیم زوج و فرد بودن عدد وارد شده توسط کاربر را بررسی کنیم.

```
clear all
clc
% Even or odd numbers
x=input('Enter your number: ');

if mod(x,2)==0
    disp('This is an Even number.')
else
    disp('This is an Odd number.')
end
```

مثال ۴.۱۰

در این مثال قصد داریم تابع سه ضابطه‌ای زیر را کدنویسی کنیم و با توجه به مقدار x که کاربر وارد می‌کند، خروجی مورد نظر را استخراج کند.

$$f(x)=\begin{cases}۱ & x<-۱\\ x^۲ & -۱\le x\le ۲\\ ۴ & x>۲\end{cases}$$

```
clear all
clc
x=input('Value of x: ');
if x<-1
    y=1;
elseif -1<=x<=2
    y=x^2;
else
    y=4;
end
disp(['The Value of function is: ',num2str(y)]);
%       x>=-1 && x<=2
```

مثال ۵.۱۰

در این مثال قصد داریم تا مجموع اعداد یک تا n را بدست آوریم:

$$\sum_{i=۱}^{n} i$$

```
clear all
clc
% This Script returns the sum of integers from 1 to n
n=input('n= ');
sum=0;
for i=1:n
```

```
    sum=sum+i;
end
disp(['The sum of ', num2str(n), ' numbers is ', num2str(sum)])
```

مثال ۶.۱۰
می‌خواهیم برنامه‌ای بنویسیم که در آن دو ماتریس وارد شده مقایسه شوند.

```
% Comparing size of two matrices
clear all
clc
A=input('First matrix: ');
B=input('Second matrix: ');
if isequal(size(A),size(B))
    disp('Two matrices are the same size')
else
    disp('Two matrices are not the same size')
end
```

مثال ۷.۱۰
در این مثال با توجه به نمره دانشجو توضیح آن در خروجی نشان داده می‌شود.

```
clear all
clc
g=input('Input your grade: ');
 switch g
   case 'A'
      fprintf('Excellent!\n');
   case 'B'
      fprintf('Well done\n');
   case 'C'
      fprintf('Well done\n');
   case 'D'
      fprintf('You passed\n');
      case 'F'
      fprintf('Better try again\n');
       otherwise
      fprintf('Invalid grade\n');
 end
```

مثال ۸.۱۰
در این مثال دنباله (الگوریتم) فیبوناچی را با استفاده از حلقه for می‌نویسیم.

$$F_n = \begin{cases} F_{n-۱} + F_{n-۲} & n > ۱ \\ ۱ & n = ۱ \\ ۰ & n = ۰ \end{cases}$$

$$\Rightarrow F = ۰, ۱, ۱, ۲, ۳, ۵, ۸, ۱۳, ۲۱, ۳۴, ۵۵, ۸۹, ۱۴۴, \ldots$$

```
clear all
clc
%%

% Fibonacci [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...]

tic

N=1000;  % Number of values in Fibonacci

f(1)=0;     % First value
f(2)=1;     % Second value
f(3)=1;     % Third value

for i=4:N
    f(i)=f(i-1)+f(i-2);
end

disp(f)
    toc
```

در ادامه مثال چند نکته را بررسی می‌کنیم:

۱. *به جای دستور disp(f) از دستور disp(f(end)) استفاده کنیم. (مقدار N را از ۱۰۰۰ به ۳۰۰۰۰ تغییر دهید.)*
مشاهده می‌کنید که اولا با این دستور تنها مقدار نهایی بردار نمایش داده خواهد شد و ثانیا این مقدار برای دنباله فیبوناچی بی‌نهایت است.

۲. *در این مرحله قبل از مقداردهی اولیه، ماتریس f را با دستور zeros تعریف کنید و زمان محاسبه را با حالت قبل مقایسه کنید. (f=zeros(1,N))*
مشاهده می‌کنید که زمان انجام محاسبات به شدت کم شده است. این مزیت تعریف ماتریس‌ها در ابتدای برنامه است.

همچنین در حالت قبل رنگ مربع سمت راست بالای صفحه برنامه نارنجی رنگ است. اگر موس را برروی آن چند لحظه نگه دارید، مشاهده می‌کنید که به شما اعلام می‌کند در برنامه شما اخطار (warning) وجود دارد. با کلیک برروی آن می‌توانید خطاها را بیابید.
دقت کنید که این مربع معمولا به سه رنگ درمی‌آید. قرمز، در این صورت در برنامه شما خطا (Error) وجود دارد. نارنجی، در این صورت در برنامه شما اخطار (Warning) وجود دارد و سبز، یعنی برنامه شما به درستی نوشته شده است و شامل هیچگونه خطا و اخطاری نیست.
با تعریف ماتریس f در ابتدای برنامه، رنگ این مربع به سبز تغییر خواهد کرد.

مثال ۹.۱۰

در این مثال برای مجموع n عدد تابع مورد نظر را می‌نویسیم.

```
% Example 9
function S = mysum (n )
```

```
% gives the sum of the first n integers
S = 0;

for i = 1: n
S = S + i;
end
```

مثال ۱۰.۱۰

در این مثال می‌خواهیم تابعی را تعریف کنیم که در آن هر بردار وارد شده به صورت افزایشی مرتب شود.

```
% Example 10
function [v] = MySort(u)
% This function sorts vector u in increasing order,
%returning the sorted vector as v

%Determine length of vector
n = length(u);

%Copy vector u to v
v = u;

%Start sorting process
    for i = 1:n-1
    for j = i+1:n
   if v(i)>v(j)
temp = v(i);
v(i) = v(j);
v(j) = temp;
end
end
end
```

مثال ۱۱.۱۰

در این مثال می‌خواهیم مجموع بر روی دو متغیر را محاسبه کنید. تابع مورد نظر به صورت زیر است:

$$\sum_{i=۱}^{n}\sum_{j=۱}^{m}\frac{۱}{(i+j)^{۲}+۱}$$

```
% Example 11
function [S] = sum2x2(n,m)

%Check if m or n are matrices
if length(n)>1 | length(m)>1
disp('sum2 - n,m must be scalar values')
else
```

```
%Calculate summation if n and m are scalars
S = 0;  %initialize sum
for i = 1:n  %sweep by index i
for j = 1:m  %sweep by index j
S = S + 1/((i+j)^2+1);
end
end
end

end
```

مثال ۱۲.۱۰

در این مثال می‌خواهیم ضرب دو ماتریس را محاسبه کنیم.

```
% Example 12
function [C] = matrixmult(A,B)

%First, check matrices compatibility
[nrA,ncA] = size(A);
[nrB,ncB] = size(B);
if ncA ~= nrB
disp('matrixmult - incompatible matrices A and B');
else
%Calculate matrix product
C = zeros(nrA,ncB);
for i = 1:nrA
for j = 1:ncB
for k = 1:ncA
C(i,j) = C(i,j) + A(i,k)*B(k,j);
end
end
end
 end
end
```