



به نام خدا

جزوه برنامه نویسی متلب، جلسه سوم

مثال شماره ۱۱ **: برنامه‌ای بنویسید که سطر N ام مثلث پاسکال را نمایش دهد.

```
example_11.m X +
1 - clearvars;
2 - prompt = 'Input N: ';
3 - N = input(prompt);
4 - for h = 0:N
5 -     A(h+1) = (factorial(N)/(factorial(h)*factorial(N-h)));
6 - end
7
8 - X = ['Pascal''s triangle : ',num2str(A)];
9 - disp(X)
10
11 % Method 2 -----
12 clearvars;
13 prompt = 'Input N: ';
14 N = input(prompt);
15 for h = 0:N
16     A(h+1) = nchoosek(N,h);
17 end
18
19 X = ['Pascal''s triangle : ',num2str(A)];
20 disp(X)
21
22 % Method 3 -----
23 clearvars;
24 prompt = 'Input N: ';
25 N = input(prompt);
26 if (N==0)|(N==1)
27     B = ones(1,N + 1);
28     X = ['Pascal''s triangle : ',num2str(B)];
29     disp(X)
30     return;
31 end
32 N = N + 1;
33 A = [1 1];
34 for h = 1 : N - 1
35     B(1) = 1; B(h+1) = 1;
36     for g = 2 : h
37         B(g) = A(g - 1) + A(g);
38     end
39     A = B;
40 end
41 X = ['Pascal''s triangle : ',num2str(A)];
42 disp(X)
```

مثال شماره ۱۲ **: برنامه‌ای بنویسید که یک آرایه را دریافت نموده با استفاده از دو حلقه تو در تو آنرا مرتب نماید.

```
example_12.m × +
1 - clearvars;
2 - prompt = 'Input array in [] ';
3 - A = input(prompt);
4 - for k = 1:length(A)
5 -     for h = k + 1:length(A)
6 -         if (A(k)>A(h))
7 -             tmp = A(k);
8 -             A(k) = A(h);
9 -             A(h) = tmp;
10 -        end
11 -    end
12 - end
13 -
14 -
15 - X = ['Sorted array : ',num2str(A)];
16 - disp(X)
17 -
18 - % Other method sort(A)
```

مثال شماره ۱۳ ***: برنامه‌ای بنویسید که یک آرایه را دریافت نموده آنرا به روش مرتب‌سازی درجی (Insertion sort) مرتب نماید.

```
example_13.m × +
1 - clearvars;
2 - prompt = 'Input array in [] ';
3 - A = input(prompt);
4 -
5 - for k = 2:length(A)
6 -     h = k - 1;
7 -     key = A(k);
8 -     while (h > 0) & (A(h)>key)
9 -         A(h + 1) = A(h);
10 -        h = h - 1;
11 -    end
12 -    A(h + 1) = key;
13 - end
14 -
15 - X = ['Sorted array : ',num2str(A)];
16 - disp(X)
17 -
18 - % Other method sort(A)
```

مثال شماره ۱۴ ***: برنامه‌ای بنویسید که یک آرایه را دریافت نموده آنرا به روش حبابی (Bubble sort) مرتب نماید.

```
example_14.m × +
1 - clearvars;
2 - prompt = 'Input array in [] ';
3 - A = input(prompt);
4 - n = length(A);
5 - while (n>0)
6 -     m = 0;
7 -     for h=2:n
8 -         if (A(h) < A(h-1))
9 -             tmp = A(h);
10 -            A(h) = A(h-1);
11 -            A(h-1) = tmp;
12 -            m = h;
13 -        end
14 -    end
15 -    n = m;
16 - end
17 -
18 -
19 - X = ['Sorted array : ',num2str(A)];
20 - disp(X)
21 -
22 - % Other method sort(A)
23 -
```

مثال شماره ۱۵ **: برنامه‌ای بنویسید که دو آرایه مرتب را دریافت نموده و آن دو آرایه را با هم ادغام و مرتب نماید.:

```

example_15.m* X +
1 - clearvars;
2 - prompt = 'Input first array in [] ';
3 - A = input(prompt);
4 - prompt = 'Input second array in [] ';
5 - B = input(prompt);
6 - n = length(A); A(n+1) = inf;
7 - m = length(B); B(m+1) = inf;
8 - C = []; h = 1; k = 1; g = 1;
9 - while (( h <= n ) || ( k <= m ))
10 -     if (A(h)<=B(k))
11 -         C(g)=A(h);
12 -         h = h + 1;
13 -         g = g + 1;
14 -     else
15 -         C(g)=B(k);
16 -         k = k + 1;
17 -         g = g + 1;
18 -     end
19 - end
20 -
21 - X = ['Sorted array : ',num2str(C)];
22 - disp(X)
23 -
24 - % Other method sort([A B])

```

مروری بر برخی توابع fix و floor و round

```

>> floor(1.55)   >> floor(-1.55)   >> fix(-1.55)   >> fix(1.55)   >> round(5.45)   >> round(5.65)   >> round(5.5)
ans =           ans =           ans =           ans =           ans =           ans =           ans =
      1           -2           -1             1             5             6             6

```

مروری بر توابع مثلثاتی و معکوس مثلثاتی

```

>> cos(3.14)   >> sin(pi/2)   >> atan(1)   >> 4*atan(1)   >> cot(pi/4)   >> sind(30)
ans =         ans =         ans =         ans =         ans =         ans =
-1.0000       1           0.7854       3.1416       1.0000       0.5000

```

سوال : تفاوت a' با $a.'$ چیست ؟

```

>> a = [ 1+i 2-3i; 2 5+3i]
a =
1.0000 + 1.0000i   2.0000 - 3.0000i
2.0000 + 0.0000i   5.0000 + 3.0000i

>> a'
ans =
1.0000 - 1.0000i   2.0000 + 0.0000i
2.0000 + 3.0000i   5.0000 - 3.0000i

>> a.'
ans =
1.0000 + 1.0000i   2.0000 + 0.0000i
2.0000 - 3.0000i   5.0000 + 3.0000i

```

```
>> A = [ 1 2 3 4]
A =
     1     2     3     4
>> sum(A)
ans =
     10
>> prod(A)
ans =
     24
```

توابع ones, zeros, rand, randc, nan

```
>> ones(2,3)
ans =
     1     1     1
     1     1     1
>> zeros(4,5)
ans =
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
>> rand(3,4)
ans =
     0.8147     0.9134     0.2785     0.9649
     0.9058     0.6324     0.5469     0.1576
     0.1270     0.0975     0.9575     0.9706
>> randc(4,3)
ans =
     0.4572    -0.0782     0.1557
    -0.0146     0.4157    -0.4643
     0.3003     0.2922     0.3491
    -0.3581     0.4595     0.4340
>> nan(4,7)
ans =
    NaN    NaN    NaN    NaN    NaN    NaN    NaN
    NaN    NaN    NaN    NaN    NaN    NaN    NaN
    NaN    NaN    NaN    NaN    NaN    NaN    NaN
    NaN    NaN    NaN    NaN    NaN    NaN    NaN
```

تفاوت عملگر دو نقطه (:) با تابع linspace

```
>> 1:1:10
ans =
     1     2     3     4     5     6     7     8     9    10
>> 1:0.5:10
ans =
Columns 1 through 12
     1.0000     1.5000     2.0000     2.5000     3.0000     3.5000     4.0000     4.5000     5.0000     5.5000     6.0000     6.5000
Columns 13 through 19
     7.0000     7.5000     8.0000     8.5000     9.0000     9.5000    10.0000
>> linspace(1,10,10)
ans =
     1     2     3     4     5     6     7     8     9    10
>> linspace(1,10,20)
ans =
Columns 1 through 12
     1.0000     1.4737     1.9474     2.4211     2.8947     3.3684     3.8421     4.3158     4.7895     5.2632     5.7368     6.2105
Columns 13 through 20
     6.6842     7.1579     7.6316     8.1053     8.5789     9.0526     9.5263    10.0000
```

اندیس بردارها

```
>> A = [ 2 4 6 8 10 12 14]
A =
     2     4     6     8    10    12    14
>> A(2:3)
ans =
     4     6
>> A(4:end)
ans =
     8    10    12    14
>> A(4:end-1)
ans =
     8    10    12
```

اندیس ماتریس

```
>> B = [ 2 3 4 5 8; 5 1 8 9 5; 1 7 3 2 4]
B =
     2     3     4     5     8
     5     1     8     9     5
     1     7     3     2     4
>> B(2:3,4:5)
ans =
     9     5
     2     4
>> B(2:3,:)
ans =
     5     1     8     9     5
     1     7     3     2     4
>> B(:,1:2)
ans =
     2     3
     5     1
     1     7
```

پیدا کردن بیشینه (ماکزیمم)، کمینه (مینیمم) و نمایه آن در بردار

```
>> A = [ 2 8 6 4 1 3 8 7 9 5 2 ]
A =
     2     8     6     4     1     3     8     7     9     5     2
>> min(A)
ans =
     1
>> max(A)
ans =
     9
>> [minVal,minInd] = min(A)
minVal =
     1
minInd =
     5
>> indx = find(A == 8)
indx =
     2     7
>> indx = find(A >= 3)
indx =
     2     3     4     6     7     8     9    10
```

تبدیل ماتریس به بردار و یافتن کوچکترین عنصر

```

A =
     1     5     6
     5     9     1
     7     5     3
     1     5     9

>> A(:)
ans =
     1
     5
     7
     1
     5
     9
     5
     5
     6
     1
     3
     9

>> A(:)'
ans =
     1     5     7     1     5     9     5     5     6     1     3     9

>> min(A(:))
ans =
     1

>> min(A)
ans =
     1     5     1

```

سوال : $\min(\min(A))$ چه چیزی را برمی گرداند. پاسخ: کوچکترین عنصر ماتریس یا بردار A

مثالهای متعددی از ضرب و تقسیم و جمع

```

>> 1 + [ 0 8 -1]
ans =
     1     9     0

>> [ 36 6 9] ./3
ans =
    12     2     3

>> [ 36 6 9] /3
ans =
    12     2     3

B =
     3     3     3

>> A ./ B
ans =
    10     3     6

>> A = [ 30, 9 , 18]; B = 3*ones(1,3);
>> B

```

مثالهایی از تقدم عملگرها

$$\begin{array}{ll}
 2 / 2 * 3 & (3) \\
 2 / 3 ^ 2 & (2/9) \\
 (2 / 3) ^ 2 & (4/9) \\
 2 + 3 * 4 - 4 & (10) \\
 2 ^ 2 * 3 / 4 + 3 & (6) \\
 2 ^ (2 * 3) / (4 + 3) & (64/7) \\
 2 * 3 + 4 & (10) \\
 2 ^ 3 ^ 2 & (64) \\
 -4 ^ 2 & (-16)
 \end{array}$$