المحاضرة الثانية: لغة البرمجة MATLAB

(The MATLAB programming language)

~~~~~~

1.المتغيرات.
2.المصفوفات والعمليات على المصفوفات.

أعداد: - م.م. محمد وليد عبد الرضا

جامعة البصرة

كلية التربية للعلوم الصرفة

قسم الرياضيات
~~~~~~

الجملة الحسابية Arithmetic Statement

الجملة الحسابية في MATLAB تكافئ المعادلة الحسابية في الجبر إلا أن MATLAB تشترط أن يكون اسم المتغير المراد حساب قيمته في الطرف الأيسر وحده بدون أشارة بينما يكون التعبير الحسابي (بقية المعادلة) في الطرف الأيمن، كما في الأمثلة التالية:

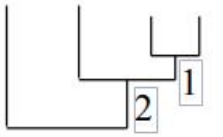
1) y = A * X + B

2) A = 3.14 * R ^ 2

مثال:

أولوية العمليات الحسابية في الجمل الحسابية:

Z = A – B / C



3

يمكن ملاحظة أن إشارة المساواة تمثل آخر أولوية حسابية بعد انتهاء جميع العمليات الحسابية في الطرف الأيمن.

(ب) المتغيرات الرمزية String Variables:

تشبه في تركيبها المتغيرات العددية والفرق الوحيد بينهما هو أن قيمة المتغير الرمزي تكون رمزية (محصورة بين علامتي اقتباس).

الجملة الرمزية String Statement

تشبه في تركيبها الجملة الحسابية والفرق الوحيد بينهما هو أن المتغير في طرفها الأيمن يكون رمزياً (محصورة بين علامتي اقتباس) والتعبير في طرفها الأيسر يكون متغير.

والأمثلة التالية توضح ذلك:

A = 'Hameed Abdul–Kareem';

N = 'Number of Student';

Dept = 'Computer Science';

ملاحظة: التعابير في الطرف الأيمن لا يكون لها قيم حسابية لو استخدمت في عمليات حسابية لأنها موضوعة داخل ' '.

الاقترانات المكتبية Library Functions:

يتوفر في معظم الحاسبات باستخدام لغة MATLAB اقترانات رياضية يكثر استعمالنا لها، مثل الدوال والاقترانات المثلثية واللوغارتيمية وغيرها ويمكن استدعائها في أي وقت، ومنها:

الاقتران	المعـــــنى
Sqrt	الجذر التربيعي
Abs	القيمة المطلقة
Exp	المرفوع إلى قوة بأساس 10
Log	اللوغاريتم الطبيعي
$\log_{10}$	اللوغاريتم العشري
$\log_{2}$	اللوغاريتم ذو الأساس 2
Sin	جيب الزاوية
Cos	جيب تمام الزاوية
Tan	ظل الزاوية
Atan	ظل معكوس الزاوية
Fix	التدوير باتجاه الصفر
Floor	التدوير باتجاه اللانهاية السالبة
Ceil	التدوير باتجاه اللانهاية الموجبة
Round	التدوير باتجاه أقرب عدد صحيح
Mod	الجزء الصحيح من حاصل القسمة
Rem	بقية القسمة
Sign	إشارة العدد إذا كانت موجبة، سالبة، صفر
Imag	القسم التخيلي
Real	القسم الحقيقي
Factor	العوامل الأولية
Isprime	يعيد true إذا كان العدد أوليا
Primes	ينشئ قائمة بالأعداد الأولية
Gcd	القاسم المشترك الأعظم
Lcm	المضاعف المشترك الأصغر

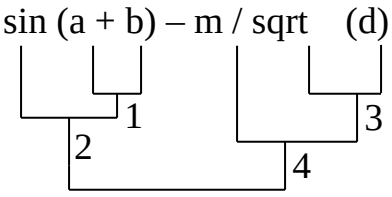
مثال:

```
>> x = 2.6;
>> y1 = fix (x); y2 = floor (x); y3 = ceil (x); y4 = round (x);
y1 = 2
y2 = 2
y3 = 3
y4 = 3
```

س/ مالفرق بين الدوال الأربعة أعلاه؟

ملاحظة:

تأخذ الاقترانات المكتبية أولوية بعد الأقواس عند تنفيذ العمليات الحسابية.



يكون تنفيذ العمليات الحسابية كما يلي:

العملية الأولى: إيجاد قيمة جمع a مع b.

العملية الثانية: إيجاد قيمة جيب الزاوية لناتج العملية (1).

العملية الثالثة: إيجاد قيمة الجذر التربيعي لـ d.

العملية الرابعة: إيجاد ناتج قيمة ناتج قسمة m على ناتج العملية (3).

العملية الخامسة: طرح ناتج العملية (4) من ناتج العملية (2) وتصبح النتيجة النهائية كمية واحدة (عدداً واحداً).

مثال: تمثل الجمل التالية إقترانات مكتبية في الجبر وإزائها قيمتها في MATLAB:

$b = \sqrt{a^2 + 10}$ ⟸ b = sqrt (a ^ 2 + 10)

z = ln (cx + ny) ⟸ z = log (c * x + n * y)

y = (sin (x + n * k)) ^ 3 ⇦ $y = \sin^3 (x + nk)$

s = atan (y / x) ⇦ $s = \tan^{-1} (y / x)$

r = 2 * sqrt (exp (x – 5)) ⇦ $r = 2\sqrt{e^{x-5}}$

t = abs (x – sqrt (y)) / (a + m) ⇦ $t = \left|x - \sqrt{y}\right| / (a + m)$

g = p ^ (3 / 2) + (a * b / c) ^ (1 / 5) ⇦ $g = p^{3/2} + \sqrt[5]{ab/c}$

المصفوفات والعمليات على المصفوفات

لقد كانت جميع الحسابات التي أجريتها حتى الآن مؤلفة من أعداد وحيدة البعد سنسميها أعداد مفردة. وتعتبر العمليات المجراة على الأعداد المفردة هي أساسيات علم الرياضيات. وبنفس الوقت، وعندما يريد الشخص إجراء نفس العملية على عدد مفرد أو أكثر، فسيحتاج إلى أكثر إعادة إجراء العملية عدة مرات، مما يعني هدر في الوقت والجهد. ولحل هذه المشكلة، عمد برنامج MATLAB إلى إجراء العمليات الرياضية على مصفوفة من البيانات.

المصفوفة البسيطة

يتعامل برنامج MATLAB مع المصفوفات بشكل مباشر وبطريقة سلسلة، إذ أن إنشاء المصفوفات يتم بطريقة سهلة جداً.

مثال (1): x = [1, 3, 7, 9, 20]

مثال (2): y = sin (x) حيث $0 \le x \le \pi$

x	0	0.1 π	0.2 π	0.3 π	0.4 π	0.5 π	0.6 π	0.7 π	0.8 π	0.9 π	π
y	0	0.31	0.59	0.81	0.95	1	0.95	0.81	0.59	0.31	0

```
x = [0  0.1 * pi  .2 * pi  .3 * pi  .4 * pi  .5 * pi  .6 * pi  .7 * pi  .8 * pi  .9 * pi
pi]
y = sin (x)
```

يقتصر كل ما عليك لإنشاء مصفوفة في لغة MATLAB على أن تبدأ بقوس يساري ثم تدخل القيم المطلوبة بفراغ أو (فارزة) ثم أغلق المصفوفة بقوس يميني. وعندما تريد كتابة sin (x) فأن برنامج MATLAB يعلم بأنك تريد حساب الجيب لكل قيم x ويقوم بوضع النتائج في مصفوفة أخرى هي y وتجعل هذه الإمكانية MATLAB مختلفة عن لغات البرمجة الأخرى.

عنونة المصفوفة أو الفهرسة

المصفوفة أعلاه تتكون من 11 عنصر، يمكن الوصول إلى أي عنصر منها باستخدام الفهرسة له.

```
>> x (3)
ans =
    0.6283
>> y (5)
```

```
ans =
    0.9511
```

ولتعريف مجموعة من العناصر بنفس الوقت فأن برنامج MATLAB يستخدم النقطتين المتعامدتين (:).

```
>> x (1: 5)
ans =
    0    0.3142    0.6283    0.9425    1.2566
```

هذه هي العناصر الخمسة الأولى من المصفوفة x، ويجبرك الرمز 1: 5 بأن تبدأ بالرقم 1 وتعدّ حتى الرقم 5.

مثال:

```
>> x (7: end)
ans =
    1.885    2.1991    2.5133    2.8274    3.1416
```

وهنا تكمل من العنصر السابع وحتى نهاية المصفوفة، إذ تشير الكلمة end إلى آخر عنصر من عناصر المصفوفة.

مثال:

```
>> y (3: -1: 1)
ans =
    0.5878    0.3090    0
```

هنا العنصر الثالث ثم الثاني ثم الأول بترتيب عكسي، ويخبرك الرمز 3: -1: 1 بأن تبدأ بالرقم 3 وتعدّ نزولاً بقيمة 1 وتقف عند الرقم 1.

مثال:

```
>> x (2: 2: 7)
ans =
    0.3142    0.9425    1.5708
```

هنا العنصر الثاني والرابع والسادس من المصفوفة x، ويخبرك الرمز 2: 2: 7 بأن تبدأ بالرقم 2 وتعدّ نحو الأعلى بـ 2 وتقف عندما تصل إلى الرقم 7.

مثال:

```
>> y ([8  2   9   1])
ans =
    0.8090    0.3090    0.5878     0
```

استخدمنا هنا مصفوفة أخرى [1 9 2 8] لوضع عناصر المصفوفة y بالترتيب الذي نرغب فيه، حيث وضع العنصر الثامن أولاً والعنصر الثاني ثانيا، بينما وضع العنصر التاسع ثالثاً والعنصر الأول رابعا. في الواقع تدل المصفوفة [1 9 2 8] عناوين العناصر المرغوبة من المصفوفة y.

مثال:

```
>> y ([1    1    3   4    2    2])
ans =
    0    0    0.5878    0.8090    0.3090    0.3090
```

مثال:

توضح الأمثلة التالية بأن برنامج MATLAB لا يقبل الدليل كرقم غير صحيح حيث يعطي رسالة خطأ.

```
>> y (3.2)
      Error
>> y (3.7)
      Error
>> y (11.6)
      Error          خطأ بسبب تجاوز الدليل طول المصفوفة
```

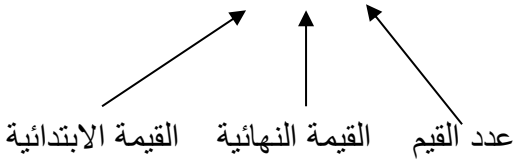
إنشاء المصفوفة

لقد قمنا سابقاً بإدخال قيم مصفوفة x عبر كتابة كل العناصر ضمن المصفوفة، وهنا الأمر مقبول لان المصفوفة x تحوي احد عشر عنصراً فقط، ماذا لو احتوت 111 عنصراً؟

هناك طريقتان لإدخال عناصر المصفوفة x، وذلك باستخدام النقطتين المتعامدتين.

أمثلة:

```
1) >>  x = (0: 0.1: 1) * pi
2) >> x = linspace (0, pi, 11 )
```



مثال:

```
>> a = [1: 7]
a =
   1   2   3   4   5   6   7
```

مثال:

```
>> b = [linspace (1, 7, 5)]
b =
   1   2.5   4   5.5   7
```

مثال:

```
>> a = (1: 7)
a =
   1   2   3   4   5   6   7
```

مثال:

```
>>  a = 1: 5    ,   b = 1: 2: 9
a =
   1   2   3   4   5
b =
   1   3   5   7   9
```

ملاحظة:

هنا تم إنشاء مصفوفتين، ولكن تذكر بأنك تستطيع دمج التعبيرين ضمن سطر واحد إذا لم تفصل بفواصل:

```
>> c = [b   a]
c =
   1   3   5   7   9   1   2   3   4   5
```

وبذلك تم إنشاء مصفوفة c مؤلفة من عناصر b متبوعة بعناصر a.

تكييف المصفوفة

بالاعتماد على المثال السابق، فان فصل العناصر بفراغات أو بفواصل عادية يحدد عناصر في أعمدة مختلفة، في حين أن استخدام الفاصلة المنقوطة يجعل العناصر واقعة في أسطر مختلفة.

مثال:

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

مصفوفة أفقية

مثال:

```
>> c = [1; 2; 3; 4; 5]
c =
   1
   2
   3
   4
   5
```

مصفوفة عمودية (كل عنصر في سطر)

مثال:

```
>> a = 1: 5
a =
   1   2   3   4   5
```

مثال:

```
>> b = a'
b =
   1
   2
   3
   4
   5
```

لقد استخدمنا هنا إشارة المنقول (المدور) لتحويل السطر a إلى العمود b.

مثال:

```
>> k = b;          (تنسيب المصفوفات)
```

مثال:

```
>> g = [1   2   3   4   ;  5   6   7   8 ]
                          ↑
                  تنزيل سطر آخر

g =
   1   2   3   4          مصفوفة متكونة من سطرين وأربعة أعمدة
   5   6   7   8
```

مثال:

```
>> g = [1   2   3  4      ↵
        5   6   7  8      ↵
        9   10  11 12]    ↵
```

كذلك فأن ضغط مفتاح Enter أو Return يخبرنا برنامج MATLAB بأن ينتقل إلى سطر جديد أثناء إدخال قيم المصفوفة.

مثال:

```
>> h = [1   2   3   ;   4   5   6   7]
      Error        عدد الأعمدة غير متساوية
```

ملاحظة:

عنصر ⟵ half = g (2, 2) ;

مصفوفة ⟵ full = g ;

مثال:

```
>> c = [1: 5; 2: 2: 10; 7: -1: 3]
c =
   1  2  3  4  5
   2  4  6  8  10
   7  6  5  4  3
>> c (1, 2)
```

```
ans =
     2
```

ملاحظة:

تقدم لغة MATLAB طريقة أخرى للإشارة إلى عناصر المصفوفة باستخدام رقم واحد فقط، ولفهم هذه الطريقة يجب التخيل بأن جميع عناصر المصفوفة مرتبة بشكل عمود واحد مكون من أعمدة المصفوفة من الأعلى إلى الأسفل (أي عناصر العمود الأول ثم الثاني ثم الثالث وهكذا).

```
>> c (12)
ans =
     4
```

العمليات الحسابية بين المصفوفة والعدد المفرد

تجري العديد من العمليات الحسابية كعملية الإضافة والطرح والضرب والقسمة بين العدد المفرد وبين جميع عناصر المصفوفة.

مثال:

```
>> g – 2          % المصفوفة g المعرفة سابقاً
ans =
    -1   0   1   2
     3   4   5   6
     7   8   9  10
```

وهنا ُطرح من كل عنصر من عناصر المصفوفة g العدد 2.

مثال:

```
>> 2 * g – 1
ans =
     1   3   5   7
     9  11  13  15
    17  19  21  23
```

أما هنا فضرب كل عنصر من عناصر المصفوفة g بالعدد 2، ثم طُرح من كل عنصر من العناصر الناتجة الرقم 1.

مثال:

```
>> 2 * g / 5 + 1
```

```
ans =
    1.4   1.8   2.2   2.6
    3     3.4   3.8   4.2
    4.6   5     5.4   5.8
```

أما في هذه الحالة، فقد ضُرب كل عنصر من عناصر المصفوفة g بالعدد 2، ثم قُسّم الناتج على العدد 5 وبعدها أُضيف لها الواحد.

العمليات الحسابية بين المصفوفات

لا تعتبر العمليات الحسابية بين المصفوفات بسيطة تماماً مثل العمليات الحسابية المجراة بين المصفوفات والأعداد المفردة. وبشكل أوضح، فالعمليات الحسابية المجراة بين مصفوفات مختلفة الأبعاد والحجوم تعد عمليات صعبة التحديد، وتعد العمليات الحسابية على المصفوفات متساوية الأبعاد من جمع وطرح وضرب وقسمة من العمليات الأساسية في لغة MATLAB واليك الأمثلة التالية:

```
>> g    % إعادة استخدام المصفوفة السابقة
g =
    1   2   3   4
    5   6   7   8
    9  10  11  12
>> h = [1  1  1  1 ; 2  2  2  2 ; 3  3  3  3]
h =
    1   1   1   1
    2   2   2   2
    3   3   3   3
>> g + h
ans =
     2   3   4   5
     7   8   9  10
    12  13  14  15
>> ans – h
```

```
ans =
     1    2    3    4
     5    6    7    8
     9   10   11   12
>> 2 * g – h
ans =
     1    3    5    7
     8   10   12   14
    15   17   19   21
>> 2 * (g – h)
ans =
     0    2    4    6
     6    8   10   12
    12   14   16   18
```

لاحظ أيضاً بأن العمليات الحسابية بين المصفوفات تعتمد نفس تسلسل أسبقية العمليات المعتمد عند إجراء العمليات الحسابية على الأعداد المفردة، ويمكن أيضاً استخدام الأقواس لكسر تلك الأولوية. كما ويمكن ضرب كل عنصر بالعنصر المناظر له من المصفوفة الأخرى أو قسمته شرط إن تُسبق إشارة الضرب أو القسمة بنقطة كما في الشكل:

```
>> g .* h
ans =
     1    2    3    4
    10   12   14   16
    27   30   33   36
```

ولقد قمنا هنا بضرب المصفوفة g بالمصفوفة h عنصر بعنصر عبر استخدام إشارة الضرب المسبوقة بنقطة.

ملاحظة: يجعل وجود النقطة أمام إشارة الضرب القياسية برنامج MATLAB يضرب المصفوفتين عنصرا بعنصر، بينما تخبر إشارة الضرب لوحدها البرنامج بان يقوم بضرب مصفوفات عادية.

```
>> g * h
    Error
```

لان عدد الأسطر للمصفوفة g ≠ عدد الأعمدة للمصفوفة h

كما إن قسمة مصفوفتين عنصراً بعنصر ممكنة عن طريق كتابة إشارة القسمة مسبوقة بنقطة كما في المثال التالي:

```
>> g ./ h
ans =
    1.0000    2.0000    3.0000    4.0000
    2.5000    3.0000    3.5000    4.0000
    3.0000    3.3333    3.6667    4.0000
```

ملاحظة: إذا سبقت إحدى إشارة القسمة بنقطة، عندها سيقوم برنامج MATLAB بتقسيم المصفوفتين عنصرا بعنصر. أما إذا كانت القسمة بدون نقطة، فإننا ستحدد قسمة مصفوفات عادية.

```
>> g .^ 2
ans =
    1    4    9    16
    25  36   49    64
    81  100  121  144
```

ولقد وجدنا هنا مربع كل عنصر من عناصر المصفوفة g.

المصفوفات القياسية

يمكّنك برنامج MATLAB من إنشاء مصفوفات قياسية، وذلك لتمتع تلك المصفوفات بخواص وميزات خاصة، وتتضمن أيضاً المصفوفات التي جميع عناصرها صفرية أو مساوية للواحد، ومصفوفات الأعداد العشوائية والمصفوفات القطرية والمصفوفات التي عناصرها أعداد ثابتة.

```
>> ones (3)          (مصفوفة واحديه)
ans =
    1  1  1
    1  1  1
    1  1  1
```

```
>> zeros (2, 5)          (مصفوفة صفرية)
```

الأسطر الأعمدة

```
ans =
     0   0   0   0
     0   0   0   0
>> size (g)              (تحديد أبعاد مصفوفة)
ans =
      3   4
>> ones (size (g))
ans =
     1   1   1   1
     1   1   1   1
     1   1   1   1
```

ملاحظة: عندما يتبع اسم المصفوفة القياسية برقم مفرد مثل ones (n) أو zeros (n) فأن برنامج MATLAB ينشيء مصفوفات مربعة n×n تحتوي على أصفارا أو واحديه على الترتيب.

```
>> eye (4)               (مصفوفة الوحدة)
ans =
     1   0   0   0
     0   1   0   0
     0   0   1   0
     0   0   0   1
>> rand (3)              (مصفوفة عشوائية)
ans =
     0.9501    0.4860    0.4565
     0.2311    0.8913    0.0185
     0.6068    0.7621    0.8214
>> rand (1, 5)
```

```
ans =
    0.4447    0.6154    0.7919    0.9218    0.7382
```

مثال:

```
>> d = pi;
>> d * ones (3, 4)
ans =
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
>> d + zeros (3, 4)
ans =
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
>> repmat (d, 3, 4)        (تكرار القيمة d بالأبعاد 3×4)
ans =
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
    3.1416    3.1416    3.1416    3.1416
```

ملاحظة: يمكن ان تكون d مصفوفة فتكون حينئذ تكرر مصفوفات وليس قيم.

التعامل مع المصفوفة

لقد امتلك برنامج MATLAB العديد من الطرق للتعامل مع المصفوفات، وكانت هذه الخاصية هي أهم مميزات البرنامج، فما إن تُحدّد المصفوفة حتى يزودك البرنامج بأقوى طرق الإدخال، التوسعة أو إعادة ترتيب بعض أجزاء المصفوفة عبر استعمال تعابير أو تعليمات محددة وممتعة، وتعتبر معرفة هذه التعليمات مفتاح الاستعمال الفعال لبرنامج MATLAB. ولشرح التعامل مع المصفوفات نأخذ الأمثلة التالية:

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

```
A =
    1   2   3
    4   5   6
    7   8   9
>> A (3, 3) = 0
A =
    1   2   3
    4   5   6
    7   8   0
```

جعل العنصر في الموقع (3, 3) صفراً.

```
>> A (2, 6) = 1
A =
    1   2   3   0   0   0
    4   5   6   0   0   1
    7   8   0   0   0   0
```

جعل العنصر في الموقع (2, 6) تكون 1 وبما ان المصفوفة A لا تمتلك ستة أعمدة، لذلك سيقوم البرنامج بتوسيعها حسب الضرورة ويضع بقي العناصر صفراً وتكون مستطيلة.

```
>> A (:, 4) = 4
A =
    1   2   3   4   0   0
    4   5   6   4   0   1
    7   8   0   4   0   0
```

جعل جميع عناصر العمود الرابع تكون 4

```
>> A (:, 4) = [4; 4; 4]
A =
    1   2   3   4   0   0
    4   5   6   4   0   1
    7   8   0   4   0   0
```

جعل جميع عناصر العمود الرابع تكون 4

```
>> A (:, 4) = [4  4  4]
    Error
```

بسبب عدم وجود فارزة منقوطة

مثال:

```
>> B = [7   8   9;  4   5   6; 1   2   3]
B =
    7   8   9
    4   5   6
    1   2   3
>> A = [1   2   3; 4   5   6; 7   8   9]
A =
    1   2   3
    4   5   6
    7   8   9
>> C = [A   B(:, [1  3])]
C =
    1   2   3   7   9
    4   5   6   4   6
    7   8   9   1   3
```

حصلنا على المصفوفة C عبر توسيع المصفوفة A بإضافة العمودين الأول والثالث من المصفوفة B.

```
>> B = A (1: 2, 2: 3)
B =
    2   3
    5   6
```

مثال: تشكيل المصفوفة B بجعل المصفوفة A كمصفوفة عمود واخذ أعمدتها عمود بعد عمود.

```
>> B = A (:)
B =
    1
    4
    7
    2
    5
```

8

3

6

9

مثال:

```
>> B = [1  2  3; 4  5  6; 7  8  9];
>> B = reshape (A, 1, 9)
B =
    1   4   7   2   5   8   3   6   9
>> B = reshape (A, [1  9]);
B =
    1   4   7   2   5   8   3   6   9
```

في المثال أعلاه إيعاز تحويل أبعاد المصفوفة الثنائية 3*3 إلى مصفوفة أحادية 9*1

مثال:

```
>> A = B
A =
    1   2   3
    4   5   6
    7   8   9
>> B (:, 2) = [ ]
B =
    1   3
    4   6
    7   9
```

تمت إعادة صياغة المصفوفة B عبر حذف كل اسطر العمود الثاني من المصفوفة B الأصلية، وعندما تضع أي عنصر مساوياً للمصفوفة الفارغة [ ]، فهذا يعني انك تريد حذفها من المصفوفة وتقليصها لتحافظ على العناصر المتبقية بعد الحذف.

مثال: إيجاد منقول (مدور) المصفوفة وإعادة تشكيلها بالتعليمة reshape.

```
>> C = B'
C =
```

```
    1   4   7
    3   6   9
>> reshape (B, 2, 3)
ans =
     1   7   6
     4   3   9
```

ملاحظة: تعمل تعليمة reshape عمود بعد عمود وللحصول على سطر بعد سطر نعمل المدور (transport).

مثال: هنا حذفنا السطر الثاني في المصفوفة C.

```
 >> C (2, :) = [ ]
C =
    1   4   7
```

مثال: استبدلنا عناصر السطر الثاني من المصفوفة A بعناصر C.

```
>> A (2, :) = C
A =
     1   2   3
     1   4   7
     7   8   9
```

مثال:

```
>> x = -3: 3
x =
   -3   -2   -1   0   1   2   3
```

هناك المصفوفات المنطقية الناتجة عن العمليات المنطقية. كما يمكن أيضاً استخدام المصفوفات المنطقية إذا كان حجمها مساويا لحجم المصفوفات المعنونة، ويتم في هذه الحالة الإبقاء على العناصر ذات القيمة (1) أي true وهي العناصر المحققة للشرط بينما يتجاهل العناصر (0) أي false وهي العناصر غير المحققة الشرط. ولنأخذ المثال التالي:

```
>> abs (x) > 1
```

```
ans =
     1   1   0   0   0   1   1
>> y = x (abs (x) > 1)
```

هنا تم إنشاء المصفوفة y من تلك العناصر من المصفوفة x التي قيمتها اكبر من الواحد.

```
y =
    -3   -2   2   3
```

ويمكن العمل مع المصفوفات الثنائية المنطقية كما عملنا مع الأحادية المنطقية، كما في المثال التالي:

```
>> B = [5   -3; 2   -4]
B =
    5   -3
    2   -4
>> x = abs (B) > 2
x =

    1   1
    0   1
>>  y = B (x)
y =
    5
    -3
    -4
```