Arrays (1A)

Young Won Lim
9/15/15

Copyright (c) 2015 Young W. Lim.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice.

Calculating the Mean of n Numbers

The mean of n numbers

$$m = \frac{\sum_{i=0}^{n-1} x_i}{n}$$

$$m = \frac{\sum_{i=0}^{4} x_i}{5} = \frac{(x_0+x_1+x_2+x_3+x_4)}{5}$$

x(**1**) x(2) x(3) x(4) x(5)

Array as a column vector

```
x = zeros(5, 1);
```

x becomes a ***5*** x ***1*** matrix

```
x = [ 1; 2; 3; 4; 5 ];
```

5 consecutive variables

index	data
1	x(1) = 0
2	x(2) = 0
3	x(3) = 0
4	x(4) = 0
5	x(5) = 0

Accessing an element

```
x(1) = 1;
x(2) = 2;
x(3) = 3;
x(4) = 4;
x(5) = 5;
```

Array as a row vector

```
x = zeros(1, 5);
```

x becomes a ***1*** x ***5*** matrix

```
x = [ 1, 2, 3, 4, 5 ];
```

5 consecutive variables

index	1	2	3	4	5
data	x(1) = 0	x(2) = 0	x(3) = 0	x(4) = 0	x(5) = 0

Accessing an element

```
x(1) = 1;
x(2) = 2;
x(3) = 3;
x(4) = 4;
x(5) = 5;
```

Range

Range Expression

base : inc : limit

returns a row vector

```
1 : 1 : 4
```

⇨ [1, 2, 3, 4]

base : limit

(inc = 1)

```
4 : -1 : 1
```

⇨ [4, 3, 2, 1]

Computing the sum of n numbers (1)

```
sum = 0;
sum = sum + x(0);
sum = sum + x(1);
sum = sum + x(2);
sum = sum + x(3);
sum = sum + x(4);
```

sum : 0;

sum : x_0

sum : $\boxed{x_0}+x_1$

sum : $\boxed{x_0+x_1}+x_2$

sum : $\boxed{x_0+x_1+x_2}+x_3$

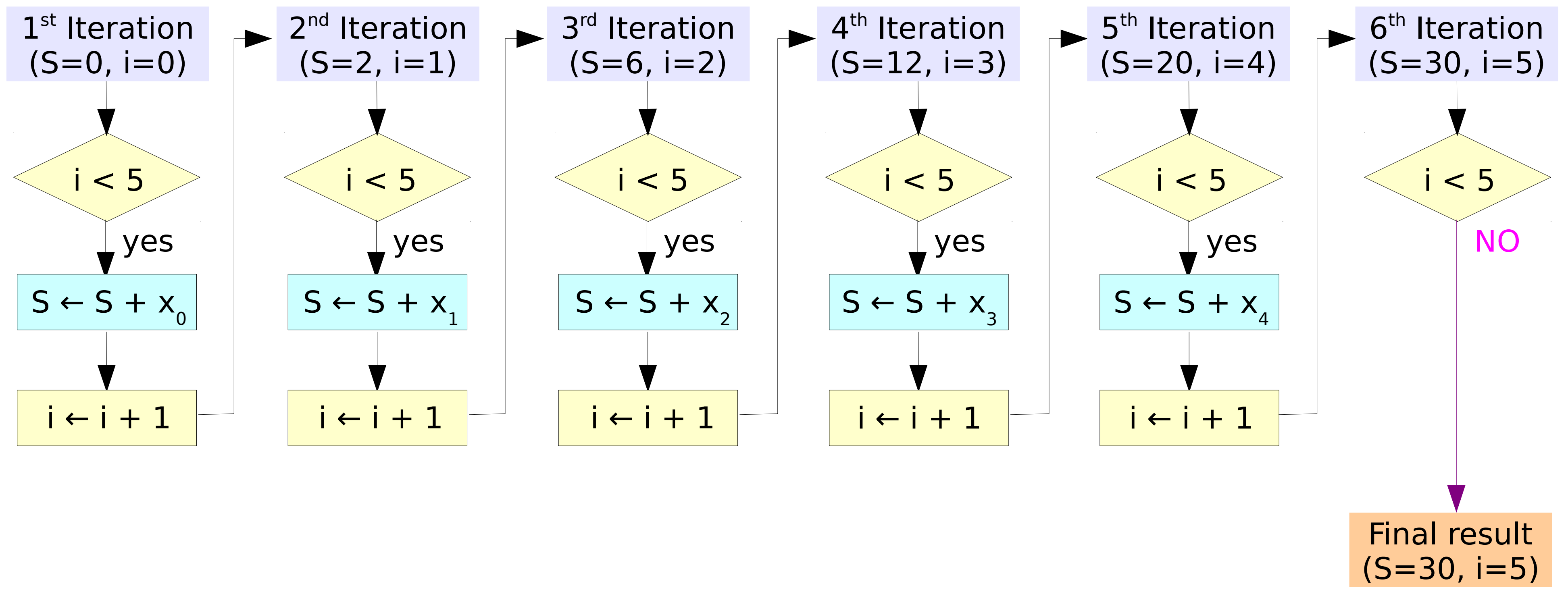
sum : $\boxed{x_0+x_1+x_2+x_3}+x_4$

```
sum = 0;
for i = 0 : 4
   sum = sum + x(i);
endfor
```

```
sum = sum(x);
```

```
sum();
prod();
cumsum();
cumprod();
```

Computing the sum of n numbers (2)



```
sum = 0;
for i = 0 : 5
   sum = sum + x(i);
endfor
```

$x_0=2$,
$x_1=4$,
$x_2=6$,
$x_3=8$,
$x_4=10$

	A	B				
i	1	0	1	2	3	4
x_i		2	4	6	8	10
S	0	2	6	12	20	30

2-D Array as a matrix

```
C = zeros(4, 4);
```

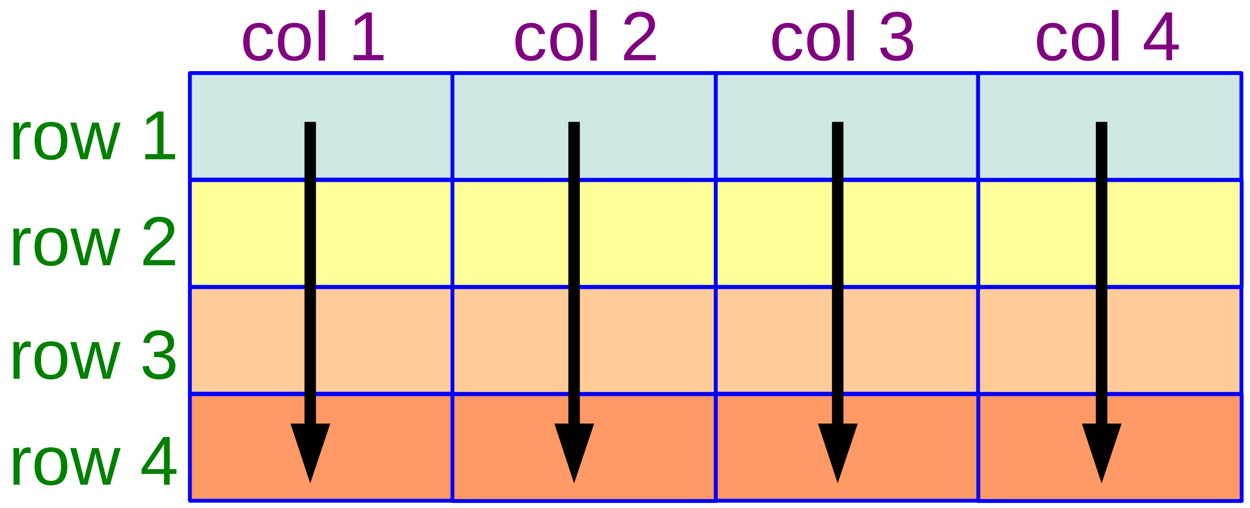
	col 1	col 2	col 3	col 4
row 1	c (1, 1)	c (1, 2)	c (1, 3)	c (1, 4)
row 2	c (2, 1)	c (2, 2)	c (2, 3)	c (2, 4)
row 3	c (3, 1)	c (3, 2)	c (3, 3)	c (3, 4)
row 4	c (4, 1)	c (4, 2)	c (4, 3)	c (4, 4)

col major ordering

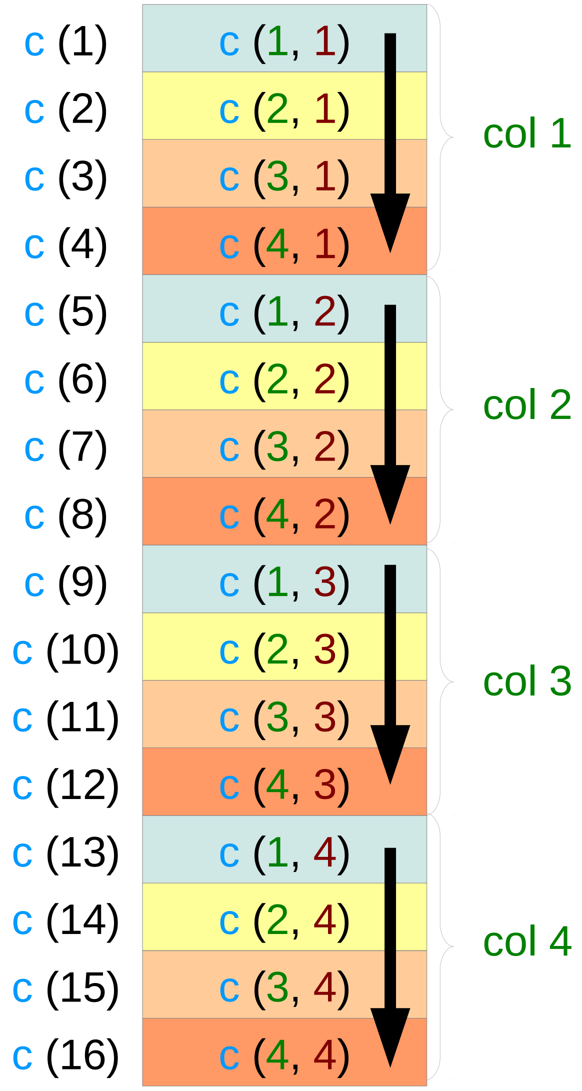
Column Major

```
C = rand(4, 4);
```

col major ordering

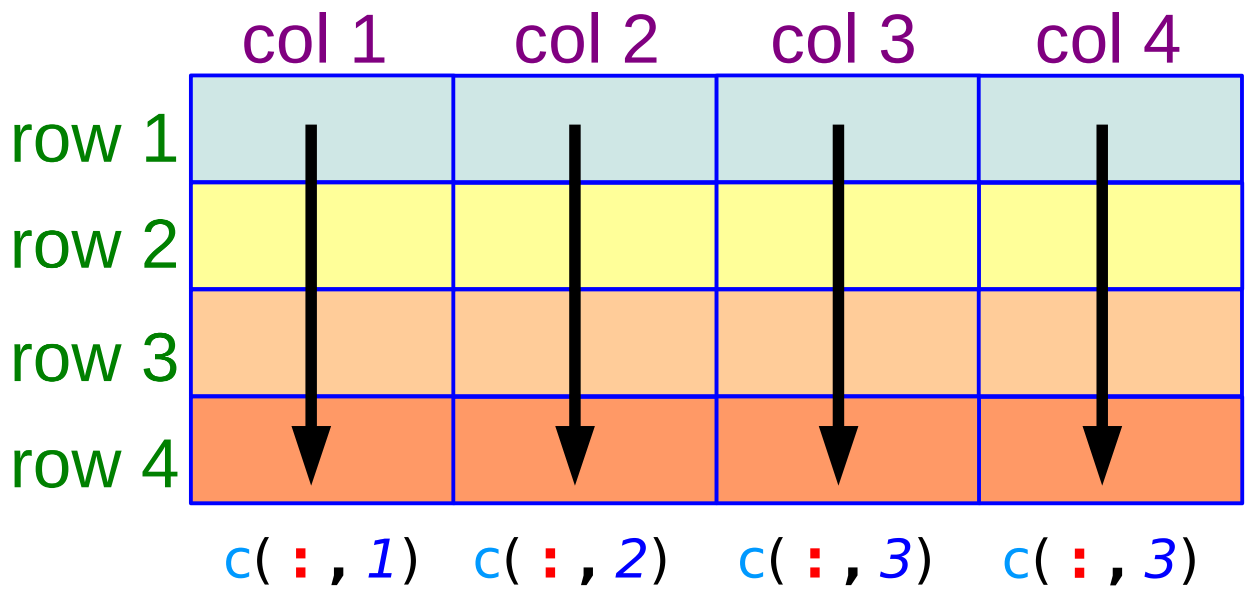


memory layout



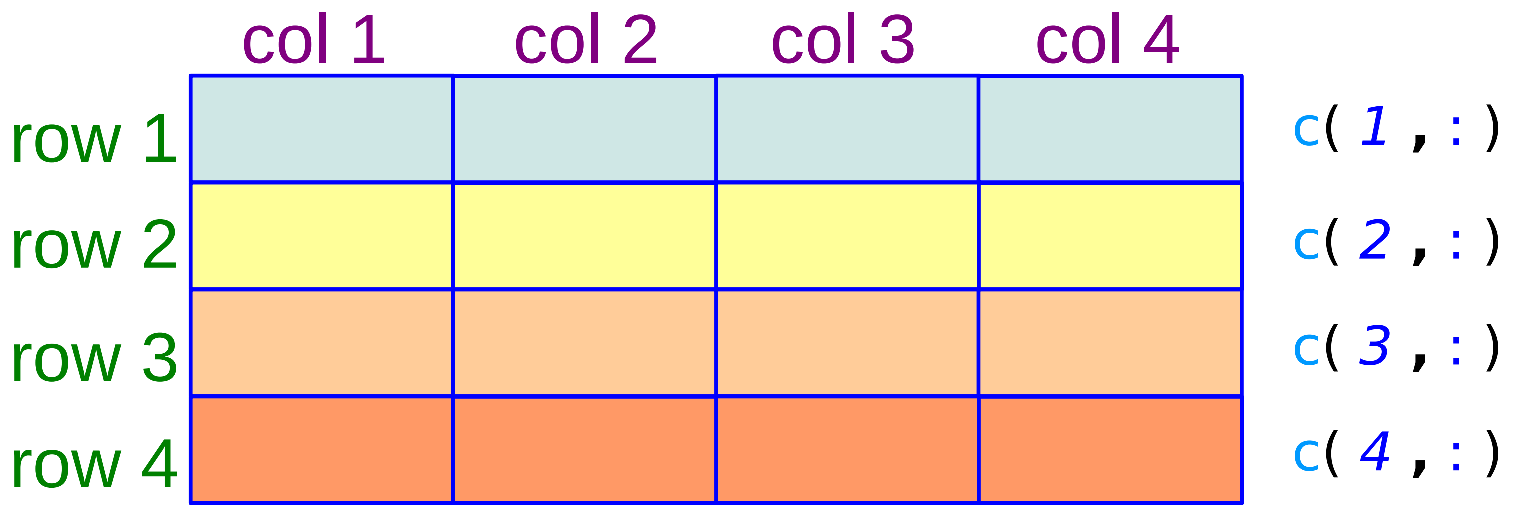
Accessing Columns

```
c( : , j )
```

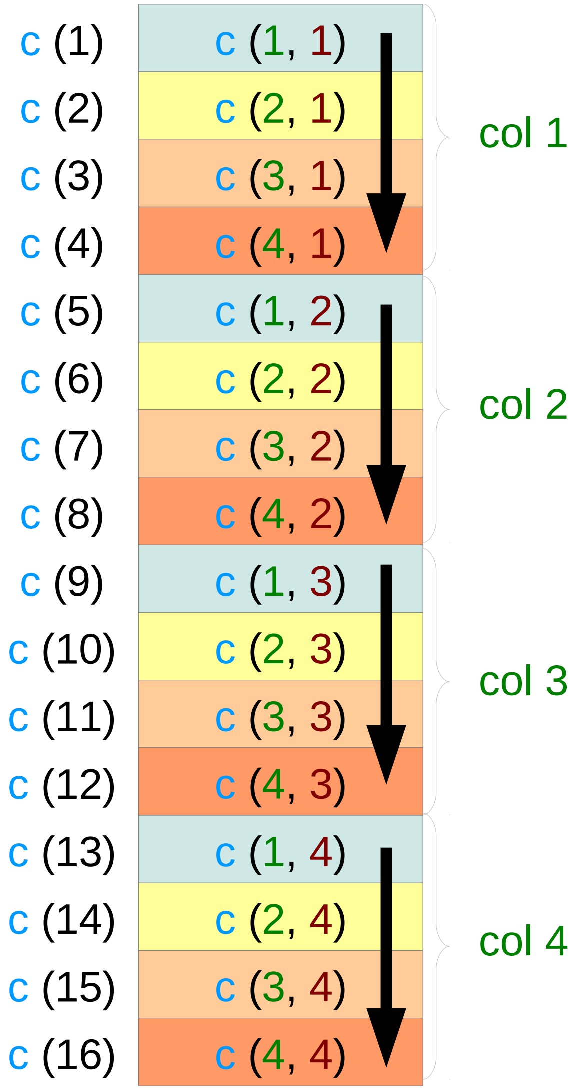


Accessing Rows

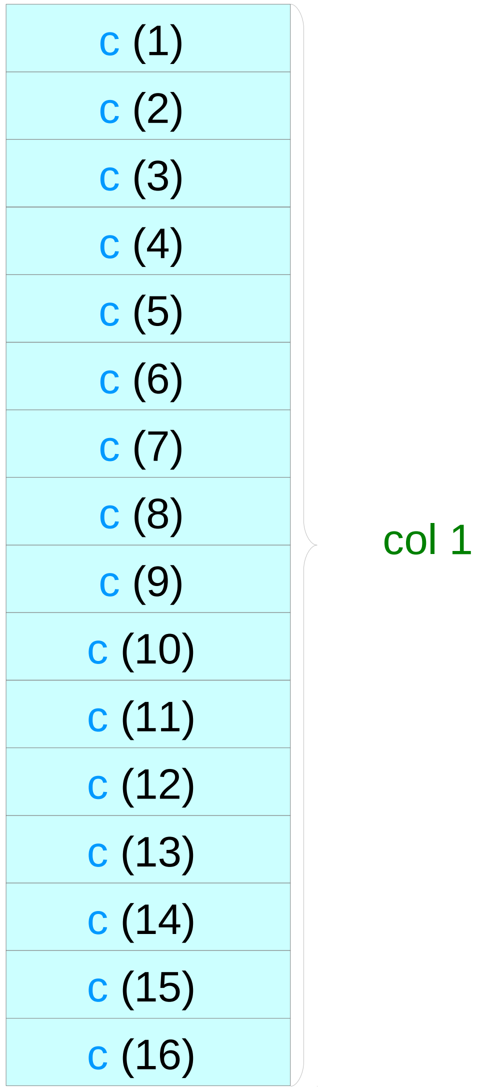
c(*i* , :)



A Single Column Vector



c(:);

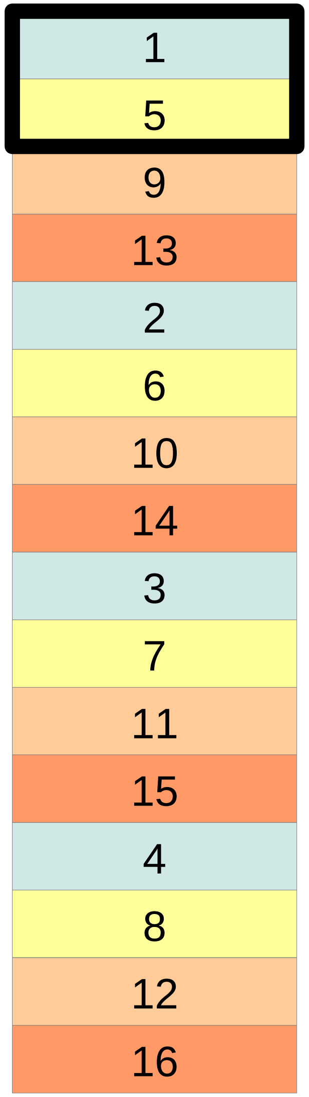


Accessing columns

a

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

a(:)



a(1) 1

a(1:2) 1 5

a([1, 2]) 1 5

a([1; 2])

1
5

Accessing Sub-matrix

a

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

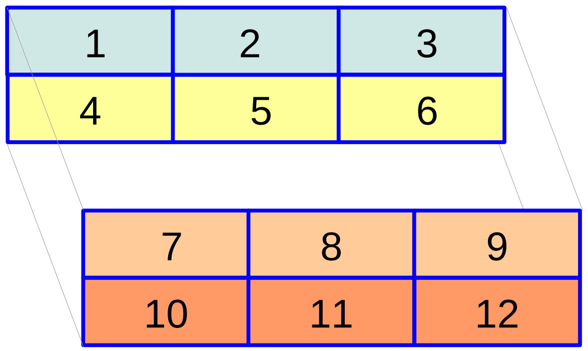
```
a( 1:2 , 1:2 )
```

1	2
5	6

```
a( : , 1:2 )
a( : , [1, 2] )
a( : , [1; 2] )
```

1	2
5	6
9	10
13	14

Multi-dimensional Array (1)



a(:, :, 1)

a(:, :, 2)

```
a( :, :, 1 ) = [ 1, 2, 3; 4, 5, 6 ];

a( :, :, 2 ) = [ 7, 8, 9; 10, 11, 12 ];
```

```
a =

ans(:,:,1) =
   1   2   3
   4   5   6

ans(:,:,2) =
    7    8    9
   10   11   12
```

Multi-dimensional Array (2)

```
b = zeros(2, 3, 2, 3);
```

```
b =

ans(:,:,1,1) =          ans(:,:,1,2) =          ans(:,:,1,3) =

   0   0   0               0   0   0               0   0   0
   0   0   0               0   0   0               0   0   0

ans(:,:,2,1) =          ans(:,:,2,2) =          ans(:,:,2,3) =

   0   0   0               0   0   0               0   0   0
   0   0   0               0   0   0               0   0   0
```

References

[1] Essential C, Nick Parlante
[2] Efficient C Programming, Mark A. Weiss
[3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
[4] C Language Express, I. K. Chun