INTRODUCTION OF LOOPS

For Loop

Example 1: based on an existing array L, create another array R with each element having the absolute value of the corresponding one in array L.

Input: L = [11 -25 32 -2 0 9 27 -91 33];

Output: R = [11 25 32 2 0 9 27 91 33];

For Loop

Before solving this problem, review how to access an array.

Basic format: *array_name*(*position*)

L = [11 -25 32 -2 0 9 27 -91 33];

… …

L(1) L(2) L(3) L(4) L(5)

For Loop

Code Example:

```
L = [11 -25 32 -2 0 9 27 -91 33];
num = length(L);

for index = 1:num
    if L(index) < 0
        R(index) = -L(index);
    else
        R(index) = L(index);
    end
end
```

Working Procedure

```
L = [11 -25 32 -2 0 9 27 -91 33];
num = length(L);

for index = 1:num
    if L(index) < 0
        R(index) = -L(index);
    else
        R(index) = L(index);
    end
end
```

Happens In Memory

L:

11	-25	32	-2	0	9	27	-91	27

num: 9

index: 1

11 < 0 ?

R:

11								

Working Procedure

```
L = [11 -25 32 -2 0 9 27 -91 33];
num = length(L);

for index = 1:num
    if L(index) < 0
        R(index) = -L(index);
    else
        R(index) = L(index);
    end
end
```

Happens In Memory

L:

11	-25	32	-2	0	9	27	-91	27

num: 9

index: 2

-25 < 0 ?

R:

11	25							

Working Procedure

```
L = [11 -25 32 -2 0 9 27 -91 33];
num = length(L);

for index = 1:num
    if L(index) < 0
        R(index) = -L(index);
    else
        R(index) = L(index);
    end
end
```

Happens In Memory

L:

11	-25	32	-2	0	9	27	-91	27

num: 9

index: 3

32 < 0 ?

R:

11	25	32						

Working Procedure

```
L = [11 -25 32 -2 0 9 27 -91 33];
num = length(L);

for index = 1:num
    if L(index) < 0
        R(index) = -L(index);
    else
        R(index) = L(index);
    end
end
```

Happens In Memory

L:

11	-25	32	-2	0	9	27	-91	27

num: 9

index: 4

-2 < 0 ?

R:

11	25	32	2					

Working Procedure

```
L = [11 -25 32 -2 0 9 27 -91 33];
num = length(L);

for index = 1:num
    if L(index) < 0
        R(index) = -L(index);
    else
        R(index) = L(index);
    end
end
```

Happens In Memory

L:

11	-25	32	-2	0	9	27	-91	27

num: 9

index: 5

0 < 0 ?

R:

11	25	32	2	0				

“for ” Loop vs “while” Loop

“for” loop – the number of iterations (rounds) is known before hand.

example 1: sum of 1, 2, 3, …, 100

example 2: traverse a vector (array)

“while” loop - the number of iterations (rounds) is unknown before hand.

example 1: get input from user

Conversion between “for ” Loop and “while” Loop

Most of the time, “for” Loop and “while” loop can be exchanged from each other.

for Loop

```
sum = 0;
data = rand(1, 100);
for index = 1:num
    sum = sum + data(index);
end
```

while Loop

```
sum = 0;
data = rand(1, 100);
index = 1;
while index < 100
    sum = sum + data(index);
    index = index + 1;
end
```

Nested “for ” Loop

Nested “for” loop is actually one “for” loop is inside another one:

```
for <condition1>
    for <condition2>

        <Executions>
        ….
    end
end
```

Nested “for ” Loop

Nested “for” loop is often used to access each element in a 2D matrix. The way of accessing a certain element in the matrix is very similar to the way of accessing an vector. The only difference is that in matrix, 2 indexes are used:

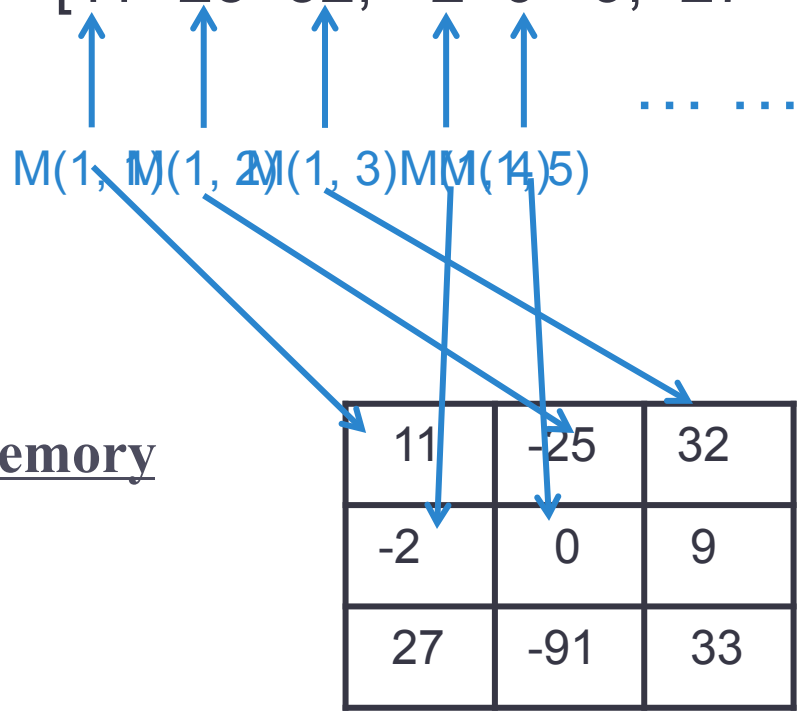
matrix_name(*index1, index2*)

Row Column

Nested “for ” Loop

Example of accessing each element in the matrix

M = [11 -25 32; -2 0 9; 27 -91 33];



Nested For Loop

Compute the sum of all the elements in a matrix

M = [3 9 2 5; 1 0 -2 4; 2 8 7 15]

```
[rows cols] = size(M);
sum = 0;
for r = 1:rows
    for c = 1:cols
        sum = sum + M(r, c);
    end
end
```

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 0

r: 1 c: 1

sum = sum + 3 = 3

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 3

r: 1 c: 2

sum = sum + 9 = 12

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 12

r: 1 c: 3

sum = sum + 2 = 14

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 14

r: 1 c: 4

sum = sum + 5 = 19

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 19

r: 2 c: 4

sum = sum + 1 = 20

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 20

r: 2 c: 2

sum = sum + 0 = 20

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 20

r: 2 c: 3

sum = sum + (-2) = 18

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 18

r: 2 c: 4

sum = sum + 4 = 22

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 22

r: 3 c: 4

sum = sum + 2 = 24

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 24

r: 3 c: 2

sum = sum + 8 = 32

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 32

r: 3 c: 3

sum = sum + 7 = 39

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
        sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 39

r: 3 c: 4

sum = sum + 15 = 54

Working Procedure

```
M = [3 9 2 5; 1 0 -2 4; 2 8 7 15];
[rows cols] = size(M);

sum = 0;
for  r  = 1:rows
    for   c = 1:cols
          sum = sum + M(r,  c);
    end
end
```

Happens In Memory

M:

3	9	2	5
1	0	-2	4
2	8	7	15

rows: 3 cols: 4

sum: 54

r: 3 c: 4

Lecture Quiz Examples

Determine the value of the Matlab variable:

```
A = 1;
for i = 1:3
        A = A - 1;
end
```

A = _____ Answer: -2

Lecture Quiz Examples

Determine the value of the Matlab variable:

```
B = 2;
while B < 2
        B = 2*B;
end
```

B = _____ Answer: 2

Lecture Quiz Examples

Determine the value of the Matlab variable:

```
for a = 1:2
        C(a) = a+1;
        C(a+1) = a+2;
end
```

C = _____

Answer: 2 3 4

Lecture Quiz Examples

Determine the value of the Matlab variable:

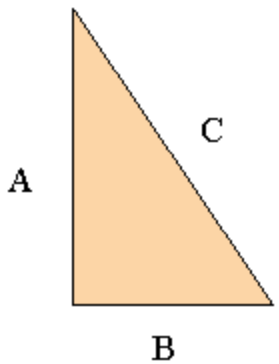
```
for r = 1:2
        for c = 1:3
                if r == c
                        F(r,c) = r;
                else
                        F(r,c) = 0;
                end
        end
end

F =     ___________
```

Answer:

1	**0**	**0**
0	2	0

Lecture Quiz Examples

Consider a right triangle as shown in Fig: 1



A	B
1	1
1	2
1	3
2	1
3	4
4	4
6	5

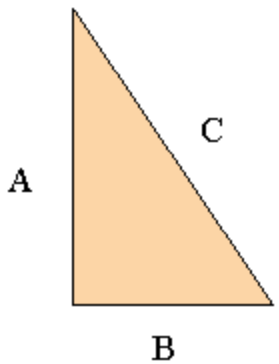
Suppose you wish to find the length of the hypotenuse C of this triangle for several conditions of side lengths A and B. The specific combinations of A and B are given in the below table:

A = [1 1 1 2 3 4 6]; or A = [1; 1; 1; 2; 3; 4; 6];

B = [1 2 3 1 4 4 5]; or B = [1; 2; 3; 1; 4; 4; 5];

Lecture Quiz Examples

Consider a right triangle as shown in Fig: 1



A	B
1	1
1	2
1	3
2	1
3	4
4	4
6	5

Suppose you wish to find the length of the hypotenuse C of this triangle for several conditions of side lengths A and B. The specific combinations of A and B are given in the above table:

```
for i= 1:7
    C(i) = sqrt(A(i)^2+B(i)^2);
end
```

Lecture Quiz Examples

Change a “for” loop into a “while” loop

```
A = [3 2 5 6 4 7 1 -2]
len = length(A);
sum = 0;
for  index  = 1:len
     sum = sum + A(index);
end
```

```
A = [3 2 5 6 4 7 1 -2]
len = length(A);
sum = 0;
index = 1;
while index <= len
     sum = sum + A(index);
     index = index + 1;
end
```

Lecture Quiz Examples

Print a matrix in a certain format.

M =

2.3	1.71	0.1	3.923
5.11	-2	3.12	0.15
2.113	10.1	9	6.2

Print matrix M so that each number has two decimal digits, and each number has a space of 8 characters. Make a new line for each row of the matrix.

```
M = [2.3 1.71 0.1 3.923; 5.11
…];
[rows cols] = size(M);
for  r  = 1:rows
     for c = 1: cols
          fprintf('%8.2f', M(r, c));
     end;
     fprintf('\n');
end
```