

CS005 Introduction to Programming: Matlab

Eamonn Keogh

eamonn@cs.ucr.edu



Golden Rules I

The only way a variable can change is if it is on the left side of an assignment

```
>> X = 25;
```

```
>> A = X + 3;
```

% Will not change X

```
>> A = TimesFive (X)
```

% Will not change X

```
>> X = X + 3;
```

% May change X

Golden Rules II

The result of any relational/
inequality test is 1 or 0

Relational/ inequality tests:

>> X == 25; % Can only be 1 or 0

==

>> X ~= Y; % Can only be 1 or 0

<

>

>> X >= max(Y); % Can only be 1 or 0

>=

<=

>> X == upper('A'); % Can only be 1 or 0

~=

```
>> upper('a') == 'A';
```

% Can only be 1 or 0, it is 1

```
>> BobsMI = 'b';
```

```
>> upper('b') == BobsMI;
```

% Can only be 1 or 0, it is 0

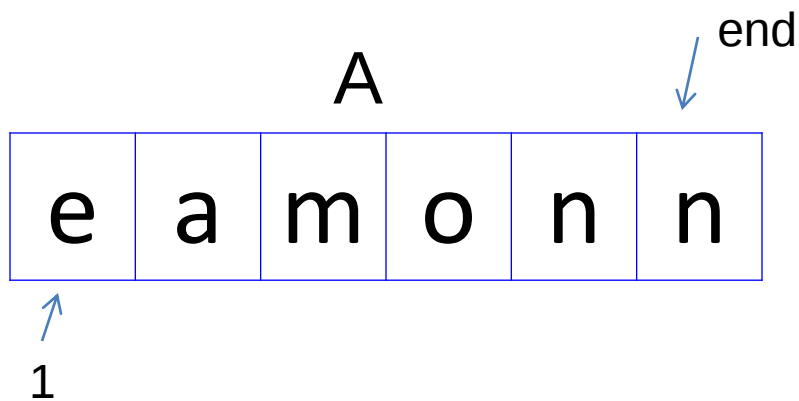
% Can only be 1 or 0

```
>> IsDrunk = GetBeersConsumed >= 2;
```

```
function IsSymmetric = BookEndString(A)
```

```
if A(1) == A(end)
    IsSymmetric = 1;
else
    IsSymmetric = 0;
end
```

Write a function
that returns true if
it begins and ends
with the same
letter



```
EDU>> BookEndString('eamonn')
ans =
    0
EDU>> BookEndString('radar')
ans =
    1
EDU>>
```

```
function IsSymmetric = BookEndString(A)
```

```
if upper(A(1)) == upper(A(end))
```

```
    IsSymmetric = 1;
```

```
else
```

```
    IsSymmetric = 0;
```

```
end
```

Fix it to be case
insensitive

```
EDU>> BookEndString('Radar')
```

```
ans =
```

```
    1
```

```
function IsRepeated = RepeatedEndLetter(A)
```

```
if A(end-1) == A(end)
```

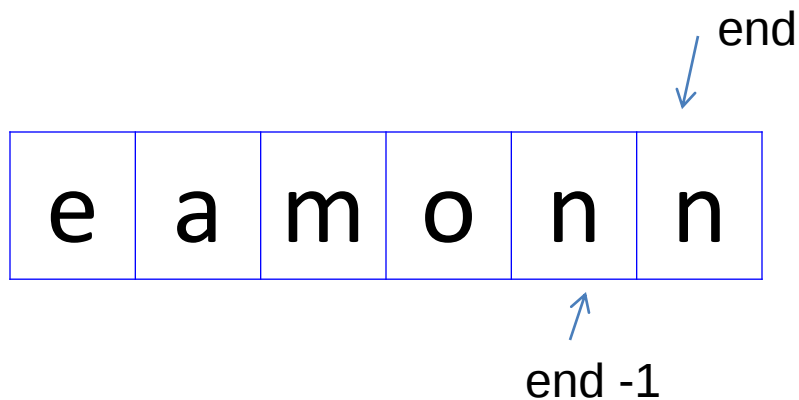
```
    IsRepeated = 1;
```

```
else
```

```
    IsRepeated = 0;
```

```
end
```

Write a function
that takes in a
string, and
returns true if the
last two letters
are the same



```
EDU>> RepeatedEndLetter('eamonn')
```

```
ans =
```

```
    1
```

```
EDU>> RepeatedEndLetter('Reese')
```

```
ans =
```

```
    0
```

```
EDU>> RepeatedEndLetter('John Hamm')
```

```
ans =
```

```
    1
```

- Alice's aunt ate apples and acorns around August.
- Becky's beagle barked and bayed, becoming bothersome for Billy.
- Carries cat clawed her couch, creating chaos.
- Dan's dog dove deep in the dam, drinking dirty water as he dove.
- Eric's eagle eats eggs, enjoying each episode of eating.
- Fred's friends fried Fritos for Friday's food.

Write a function
that takes in a
string that
contains two or
more words, and
returns true if the
first two words
are alliterative

T	o		t	e	s	t
---	---	--	---	---	---	---

```
function SecondWordOnwards = StripFirstWord(A)
```

```
for i = 1 : length(A)-1
```

```
    if A(i) == ' '
```

```
        StartOfSecondWord = i + 1;
```

```
    end
```

```
end
```

```
SecondWordOnwards = A(StartOfSecondWord:end);
```

T	o		t	e	s	t
---	---	--	---	---	---	---

```
EDU>> StripFirstWord('Dr. Keogh')  
ans =  
Keogh
```

Looks good... but
there is a bug..

```
function SecondWordOnwards = StripFirstWord(A)
```

```
for i = 1 : length(A)-1
```

```
    if A(i) == ' '
```

```
        StartOfSecondWord = i + 1;
```

```
    end
```

```
end
```

```
SecondWordOnwards = A(StartOfSecondWord:end);
```

```
EDU>> StripFirstWord('live and let die')
```

```
ans =
```

```
die
```

We don't start the second word at the *first* space, we start at the *last* space

```
function SecondWordOnwards = StripFirstWord(A)
```

```
for i = 1 : length(A)-1
```

```
    if A(i) == ' '
```

```
        StartOfSecondWord = i + 1;
```

```
        break;
```

```
    end
```

```
end
```

```
SecondWordOnwards = A(StartOfSecondWord:end);
```

Fixed!

```
EDU>> StripFirstWord('live and let die')
```

```
ans =
```

```
and let die
```

`function Alliterative = IsAlliterative(A)` →

T	o		t	e	s	t
---	---	--	---	---	---	---

`A = upper(A);` →

T	O		T	E	S	T
---	---	--	---	---	---	---

`FirstLetterFirstWord = A(1);` →

T

`A = StripFirstWord(A);` →

T	E	S	T
---	---	---	---

`FirstLetterSecondWord = A(1);` →

T

`if FirstLetterFirstWord == FirstLetterSecondWord`

T

 ==

T

`Alliterative = 1;`

`else`

`Alliterative = 0;`

`end`

```
EDU>> IsAlliterative('To test')
ans =
     1
EDU>> IsAlliterative('To rest')
ans =
     0
```

```
EDU>> StringRotation('eamonn')
```

eamonn

amonne

monnea

onneam

nneamo

neamon

eamonn

Write a function
that takes in a
string and
displays it, then
every rotation of
it

e	a	m	o	n	n
---	---	---	---	---	---

EDU>> StringRotation('eamonn')

eamonn

amonne

monnea

onneam

nneamo

neamon

eamonn



Amonne A



NEAMON

Monnea decora is a species of beetle in the family Carabidae,

```
EDU>> StringRotation('eamonn')  
eamonn  
amonne
```

Write a function that
takes in a string and
displays it, then
every rotation of it

A =

e	a	m	o	n	n
---	---	---	---	---	---

A =

e	a	m	o	n	n	e	a	m	o	n	n
a						m	o	n	n	e	

A = [A(2:end) A(1)];

```
function dummy = StringRotation(A)
```

```
    disp(A)
```

```
    for i = 1 : length(A)
```

```
        A = [A(2:end) A(1)];
```

```
        disp(A)
```

```
    end
```