CS005 Introduction to Programming: Matlab

Eamonn Keogh eamonn@cs.ucr.edu



ABSTINENCE

It's 100% 99.999% effective.

Searching Arrays

- Many problems involve searching an array (including a string) for special values.
- Examples include
 - Searching an array for the largest value (which player in my team makes the most money?)
 - Searching an array for a particular value (do I have a student with the ID 8932328?)
 - Searching a string for a particular value(s) (Does this string "control" have vowels? CNTRL)

Searching Arrays

- Many problems involve searching an array (including a string) for special values.
- These problems are so important that Matlab has special build-in functions to solve some of them.
- We have seen:

```
EDU>> max([ 2 3 5 3 1 6 4 3 8])
ans =
```

```
8
```

 However, let us write some of these function from scratch to build our skill set...

Finding Existence of a Given Number

- EDU>> TeamAges = [5 , 6, 11, 9 , 8];
- EDU>> AgeToFind = 9;
- EDU>> EamonnsFindNumber(TeamAges,AgeToFind)
- ans =
- 1 EDU>> EamonnsFindNumber(TeamAges,77)
- ans =

0

EDU>> EamonnsFindNumber([66, 77, 45],77)

ans =

1

Let us write a function that takes in an array, a number to find in that array, and returns true/false if the number is in the array

[4, 2, 9] 2

function TargetNumberExists = EamonnsFindNumber(ArrayToSearch,TargetNum)

```
TargetNumberExists = 0; % Assume the number does not exist for now
```

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

```
if ArrayToSearch(i) == TargetNum
  TargetNumberExists = 1; % Change our assumption
end
```

end



function TargetNumberExists = EamonnsFindNumber(ArrayToSearch,TargetNum)

```
TargetNumberExists = 0; % Assume the number does not exist for now
```

for i = 1 : length(ArrayToSearch)

```
if ArrayToSearch(i) == TargetNum
  disp(['the if statement was triggered with i equals ', num2str(i)])
  TargetNumberExists = 1; % Change our assumption
  end
```

end

EDU>> EamonnsFindNumber([66, 77, 45],77) the if statement was triggered with i equals 2 ans = 1 Useful hint: It is good idea to add temporary code to echo what is happening in the code. This is especially true as we consider more complex code

Finding Location of Given Number

- EDU>> TeamAges = [5 , 6, 11, 9 , 8];
- EDU>> AgeToFind = 9;
- EDU>> FindNumberLocation(TeamAges,AgeToFind)
- ans =
 - 4
- EDU>> FindNumberLocation(TeamAges,77)
- ans =
 - NaN

- Let us write a function that takes in an array, a number to find in that array, and returns the location of that number in the array.
- If the number is not in the array, it return NaN



Finding Location of Given Number

Important note:

The previous slide did not specify what to do if the target number appears more than once. It is our job to clarify these things We could:

- Report the first occurrence
- Report the second occurrence
- Do something else

Let us write a function that takes in an array, a number to find in that array, and returns the location of that number in the array.

If the number is not in the array, it return NaN



function NumLocation = FindNumberLocation(ArrayToSearch,TargetNum)

NumLocation = NaN; % Assume the number does not exist for now

```
for i = 1 : length(ArrayToSearch)
    if ArrayToSearch(i) == TargetNum
        NumLocation = i; % Assumption was wrong, record the location
        end
    end
end
```

```
EDU>> TeamAges = [ 5 , 6, 5, 9 , 7];
EDU>> FindNumberLocation(TeamAges,5)
ans =
3
```

function NumLocation = FindNumberLocation(ArrayToSearch,TargetNum)

NumLocation = NaN; % Assume the number does not exist for now

```
for i = 1 : length(ArrayToSearch)
    if ArrayToSearch(i) == TargetNum
        NumLocation = i; % Assumption was wrong, record the location
        end
    end
end
```

Does our code find the first or last occurrence of a number?

It finds the *last* occurrence

In the next slide we will see how to find the *first* instead EDU>> TeamAges = [5 , 6, 5, 9 , 7]; EDU>> FindNumberLocation(TeamAges,5) ans = 3 function NumLocation = FindNumberLocation(ArrayToSearch,TargetNum)

NumLocation = NaN; % Assume the number does not exist for now

```
for i = 1 : length(ArrayToSearch)
    if ArrayToSearch(i) == TargetNum
        NumLocation = i; % Assumption was wrong, record the location
        break;
    end
end
```

The matlab command break terminates a for or while loop. The moment that Matlab encounters a break it jumps to the end of the loop and continues with the rest of the program...

EDU>> TeamAges = [5 , 6, 5, 9 , 7]; EDU>> FindNumberLocation(TeamAges,5) ans = 1

Finding Vowel Count

Let us write a function that takes in a string, and returns a count of how many vowels it has.

Let us use our strategy of solving a simpler problem and working our way upto the final solution...

Let us use our strategy of echoing to the screen so we can debug the code..



function VowelCount = FindVowels(StringToSearch) VowelCount = 0; % Assume there are no vowels for i = 1 : length(StringToSearch) if StringToSearch(i) == 'a' VowelCount = VowelCount + 1; disp(['An a was found when i was ', num2str(i)]) end end

> EDU>> FindVowels('eamonn') An a was found when i was 2 ans = 1 EDU>> FindVowels('abba') An a was found when i was 1 An a was found when i was 4 ans = 2

function VowelCount = FindVowels(StringToSearch) VowelCount = 0; % Assume there are no vowels for i = 1 : length(StringToSearch) if StringToSearch(i) == 'a' VowelCount = VowelCount + 1; disp(['An a was found when i was ', num2str(i)]) end end

There is a bug! We are not finding uppercase 'A'

EDU>> FindVowels('ABBA') ans = 0 function VowelCount = FindVowels(StringToSearch) VowelCount = 0; % Assume there are no vowels for i = 1 : length(StringToSearch) if lower(StringToSearch(i)) == 'a' VowelCount = VowelCount + 1; disp(['An a was found when i was ', num2str(i)]) end end

There is a simple fix

EDU>> FindVowels('ABBA') An a was found when i was 1 An a was found when i was 4 ans = function VowelCount = FindVowels(StringToSearch)

```
VowelCount = 0; % Assume there are no vowels
```

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

```
if lower(StringToSearch(i)) == 'a'
    VowelCount = VowelCount + 1;
    disp(['An a was found when i was ', num2str(i)])
end
```

```
if lower(StringToSearch(i)) == 'e'
    VowelCount = VowelCount + 1;
    disp(['An e was found when i was ', num2str(i)])
end
```

end

We can begin to add more code..

EDU>> FindVowels('eamonn') An e was found when i was 1 An a was found when i was 2 ans = 2

Finding The Largest Value

- Let us write a function that finds the largest number in an array.
- As it happens, the built in max function does this, but we need to sharpen our skills...



function LargestNumber = EamonnsMax(InputArray)

LargestNumber = -Inf; % Assume this is the largest number

for i = 1 : length(InputArray)
 if InputArray(i) > LargestNumber
 disp('I need to update the largest number')
 LargestNumber = InputArray(i);
 end
end
EDU>> EamonnsMaxe
InputArray(i)

1	4	2	8	4
---	---	---	---	---

EDU>> EamonnsMax([1 4 2 8 4]) I need to update the largest number I need to update the largest number I need to update the largest number ans =

8

4

EDU>> EamonnsMax([4 3 2 1]) I need to update the largest number ans =