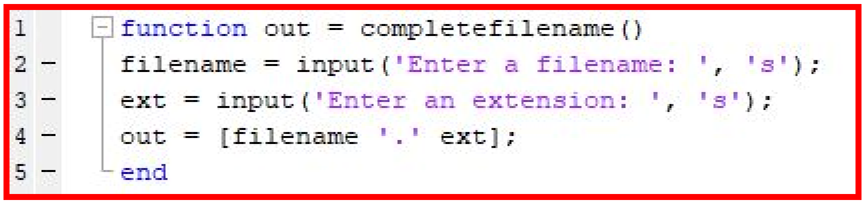
ENG EK 127 - Worksheet 7

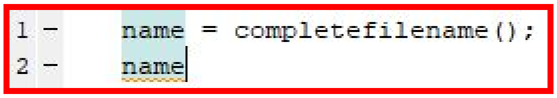
Name: Ande Chen Section: C1

The minimum set is all problems.

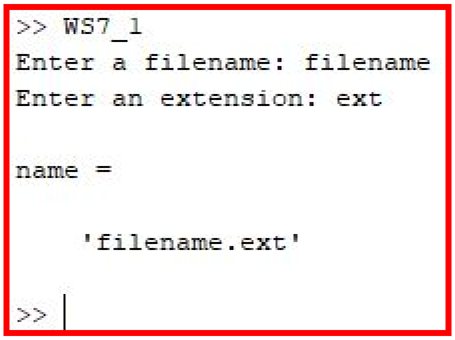
1) Write a function that will prompt the user separately for a filename and extension and will create and return a character vector with the form ‘filename.ext’.



```
function out = completefilename()
filename = input('Enter a filename: ', 's');
ext = input('Enter an extension: ', 's');
out = [filename '.' ext];
end
```



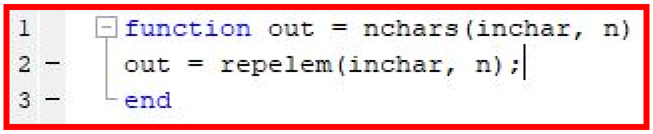
```
name = completefilename();
name
```



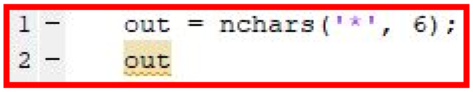
```
>> WS7_1
Enter a filename: filename
Enter an extension: ext

name =

    'filename.ext'

>>
```

2) Write a function *nchars* that will create a character vector of n characters, without using any loops or selection statements.

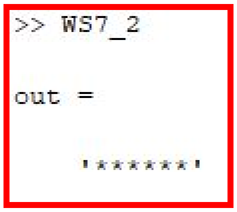
```
>> nchars('*', 6)
ans =
          '******'
```



```
function out = nchars(inchar, n)
out = repelem(inchar, n);
end
```



```
out = nchars('*', 6);
out
```

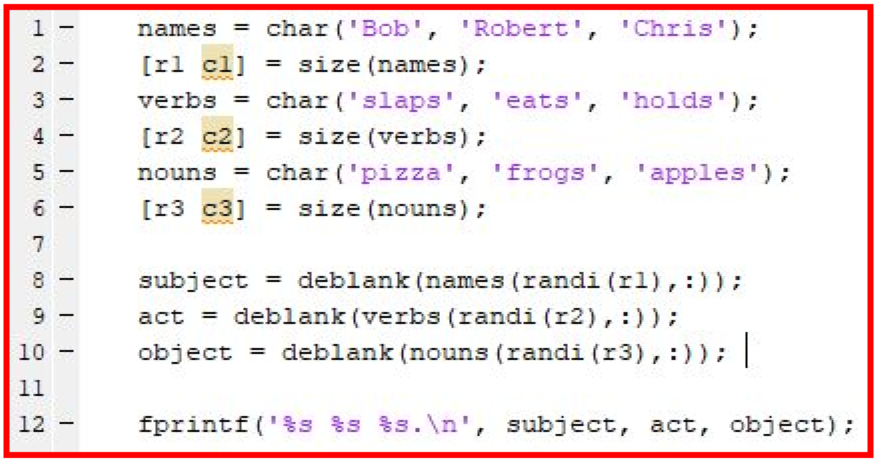


```
>> WS7_2

out =

    '******'
```

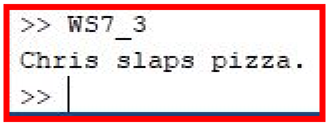
3) Create three char matrix variables that store people's names, verbs, and nouns. For example,

```
>> names = char('Harry','Xavier','Sue');
>> verbs = char('loves','eats');
>> nouns = char('baseballs','rocks','sushi');
```

Write a script that will initialize these variables, and then print sentences using one random element from each (e.g. 'Xavier eats sushi').



```
names = char('Bob', 'Robert', 'Chris');
[r1 c1] = size(names);
verbs = char('slaps', 'eats', 'holds');
[r2 c2] = size(verbs);
nouns = char('pizza', 'frogs', 'apples');
[r3 c3] = size(nouns);

subject = deblank(names(randi(r1),:));
act = deblank(verbs(randi(r2),:));
object = deblank(nouns(randi(r3),:));

fprintf('%s %s %s.\n', subject, act, object);
```



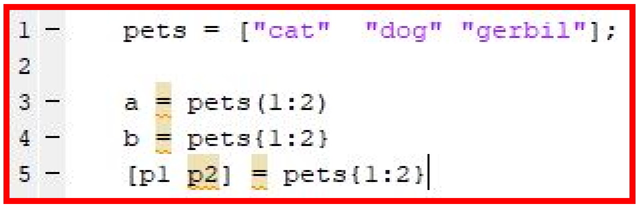
```
>> WS7_3
Chris slaps pizza.
>>
```

4) Create a string array that contains pet types, e.g.,

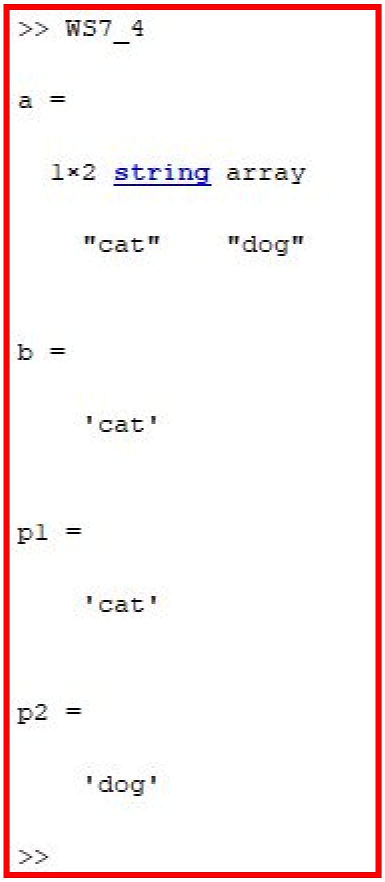
```
pets = ["cat"  "dog" "gerbil"];
```

Show the difference in the following methods of indexing into the first two strings:

```
pets(1:2)
pets{1:2}
[p1 p2] = pets{1:2}
```

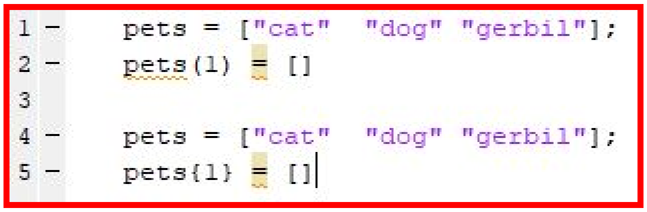


```
pets = ["cat"  "dog" "gerbil"];

a = pets(1:2)
b = pets{1:2}
[p1 p2] = pets{1:2}
```

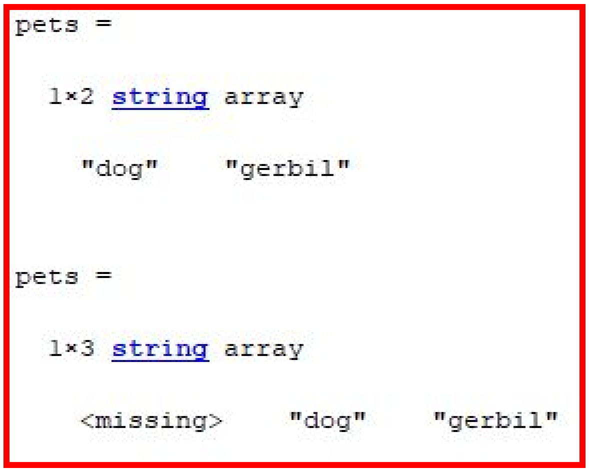


```
>> WS7_4

a =

  1×2 string array

    "cat"    "dog"


b =

    'cat'


p1 =

    'cat'


p2 =

    'dog'

>>
```

5) Show the difference between assigning an empty vector to an element in a string array, by using parentheses and by using curly braces to index into the element.

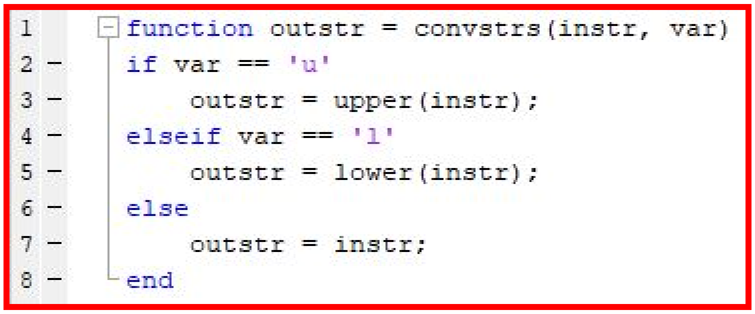


```
pets = ["cat"  "dog" "gerbil"];
pets(1) = []

pets = ["cat"  "dog" "gerbil"];
pets{1} = []
```

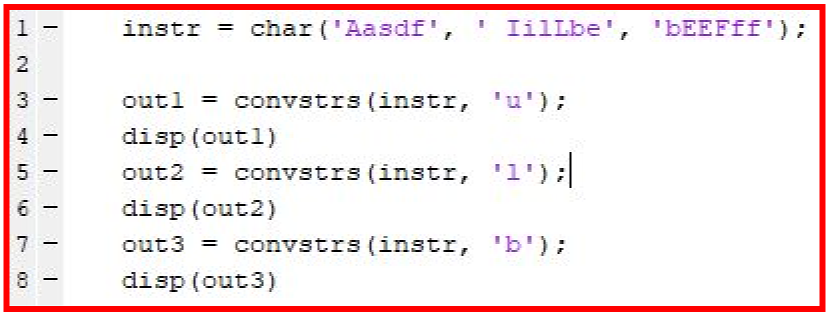


```
pets =

  1×2 string array

    "dog"    "gerbil"


pets =

  1×3 string array

    <missing>    "dog"    "gerbil"
```

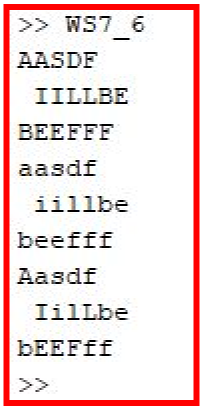
6) Write a function *convstrs* that will receive a char matrix of characters and a character ‘u’ or ‘l’. If the character is ‘u’, it will return a new char matrix with all of the characters in uppercase. If the character is ‘l’, it will return a new char matrix with all of the characters in lowercase. If the character is neither ‘u’ nor ‘l’, the char matrix that is returned will be identical to the input char matrix.



```
function outstr = convstrs(instr, var)
if var == 'u'
    outstr = upper(instr);
elseif var == 'l'
    outstr = lower(instr);
else
    outstr = instr;
end
```



```
instr = char('Aasdf', ' IilLbe', 'bEEFff');

out1 = convstrs(instr, 'u');
disp(out1)
out2 = convstrs(instr, 'l');
disp(out2)
out3 = convstrs(instr, 'b');
disp(out3)
```



```
>> WS7_6
AASDF
 IILLBE
BEEFFF
aasdf
 iillbe
beefff
Aasdf
 IilLbe
bEEFff
>>
```