Computer Programming

Bachelor in Biomedical Engineering

Bachelor in Applied Mathematics and Computing

Course 2020 / 2021

Exercise Sheet 8

Strings

– SOLUTIONS –

Content Table

Exercise 1

Write a program that asks the user to introduce words/sentences until he/she introduces a blank (empty) word. Then, the program prints all the words concatenated using the symbol ‘-‘.

Example of execution:

```
Introduce a word: Computer
Introduce a word: Programming
Introduce a word: Biomedical
Introduce a word:
Computer-Programming-Biomedical
```

SOLUTION

```
word = input('Introduce a word: ', 's');
mysentence = word;
while (isempty(word)==0)
    word = input('Introduce a word: ', 's');
    if (isempty(word)==0)
        mysentence = [mysentence '-' word];
    end
end
fprintf('%s\n', mysentence);
```

Exercise 2

Same exercise, but the words should now appear concatenated in reverse introduction order.

Example of execution:

```
Introduce a word: Computer
Introduce a word: Programming
Introduce a word: Biomedical
Introduce a word:
Biomedical-Programming-Computer
```

SOLUTION

```
word = input('Introduce a word: ', 's');
mysentence = word;
while (isempty(word)==0)
    word = input('Introduce a word: ', 's');
    if (isempty(word)==0)
        mysentence = [word '-' mysentence];
    end
end
fprintf('%s\n', mysentence);
```

Exercise 3

Write a program that asks the user to introduce a sentence, and prints the words the sentence contains on screen.

Example of execution:

```
Introduce a sentence: A cat is an animal
The words in the sentence are:
A
cat
is
an
animal
```

SOLUTION

```
sentence = input('Introduce a sentence: ', 's');
while (isempty(sentence)==0)
    [word, remain] = strtok(sentence, ' '); % or strtok(sentence)
    fprintf('%s\n', word);
    sentence = remain;
end
```

Exercise 4

Write a program that asks the user to introduce his/her names and surnames (all together), and prints the initials.

Example of execution:

```
Introduce your name and surname: Bill Gates Garcia
Your initials are B. G. G.
```

SOLUTION

```
initials = [];
names = input ('Introduce names and surnames: ', 's');
while (isempty(names)==0)
    [word, remain] = strtok(names, ' ');
    initials = [initials ' ' upper(word(1)) '.'];
    names = remain;
end
fprintf('Your initials are %s\n', initials);
```

Exercise 5

Write a program that asks the user to introduce a sentence, and prints the number of vowels in the sentence on screen.

Example of execution:

```
Introduce a sentence: The cat is on the table
There are 7 vowels in the sentence.
```

SOLUTION

```
sentence = input('Introduce a sentence: ','s');
vowels = ['a' 'e' 'i' 'o' 'u' 'A' 'E' 'I' 'O' 'U'];
count = 0;
for v = vowels
    vpos = strfind(sentence, v); % check how many times each vowel appears
    count = count + length(vpos);
end
fprintf('There are %d vowels in the sentence\n', count);
```

ANOTHER SOLUTION

```
sentence = input('Introduce a sentence: ','s');
vowels = ['a' 'e' 'i' 'o' 'u' 'A' 'E' 'I' 'O' 'U'];
count = 0;
for letter = sentence
    if contains(vowels, letter) % check for each letter whether they are
present in the vector 'vowels'
        count = count + 1;
    end
end
fprintf('There are %d vowels in the sentence\n', count);
```

Exercise 6

Create a function *rmChars* that receives a string and a character, and returns the same string without any occurrence of the specified character.

Example of execution (using the command window):

```
>> rmChars('Cats are my favorite animal','a')

ans = 'Cts re my fvorite niml'
```

SOLUTION

```
function [outstring] = rmChars (instring, inchar)
% rmChars receives a string (instring) and a char (inchar)
% and returns the string (outstring) without any occurrences of the
character
outstring = [];
for i=1:length(instring)
    if (instring(i) ~= inchar)
        outstring = [outstring instring(i)];
    end
end
end
```

Exercise 7

Write a function that receives a sentence and returns the sentence without any text within parentheses.
Write a program that asks the user to introduce a sentence and uses the function to produce a new sentence without text within parentheses.

Example of execution:

```
Introduce a sentence: The cat (which is an animal) is
a natural predator
The cat is a natural predator
```

Note: You can consider that a sentence can only contain one piece of text within parentheses.

FUNCTION

```
function [outphrase] = removeParenthesis (inphrase)
% part1 = part before (
% part2 = part after and including (
[part1, part2] = strtok(inphrase, '(');
% part 3 = part before )
% part 4 = part after and including )
[part3, part4] = strtok(part2, ')');
part4(1) = ''; % remove the ) character at the beginning of part4
% put together the part before the parentheses are opened (part1),
% and the part after the parentheses are closed (part4)
outphrase = [part1 part4];
end
```

FUNCTION: ANOTHER SOLUTION

```
function [outphrase] = removeParenthesis (inphrase)
outphrase = [];
posopen = strfind(inphrase, '(' );
posclose = strfind(inphrase, ')' );
if (isempty(posopen)==0) && (isempty(posclose)==0)
    if posopen(1) < posclose(1)
        outphrase = [inphrase(1:posopen-1)
inphrase(posclose+1:length(inphrase))];
    end
end
end
```

MAIN PROGRAM

```
st = input('Introduce a sentence: ','s');
outst = removeParenthesis(st);
fprintf('%s\n', outst);
```

Exercise 8

Write a program that asks the user to introduce a sentence, and prints the words the sentence contains on screen. **Solve the problem without using *strtok.***

Example of execution:

```
Introduce a sentence: A cat is an animal
The words in the sentence are:
A
cat
is
an
animal
```

TIP TO SOLVE THE EXERCISE:

STEP 1: ask the user to introduce a sentence

STEP 2: find the positions of the spaces in the sentence

STEP 3: extract the words in between the spaces

SOLUTION

```
sentence = input('Introduce a sentence: ', 's');
whitespaces = strfind(sentence, ' ');
wini = 1; % start position word
for i=1:length(whitespaces)
    wend = whitespaces(i) - 1; % end position word
    word = sentence(wini : wend);
    fprintf('%s\n', word);
    wini = whitespaces(i) + 1;
end
word = sentence(wini : length(sentence)); % necessary for last word
fprintf('%s\n',word);
```