

Computer Programming
Bachelor in Biomedical Engineering
Bachelor in Applied Mathematics and Computing
Course 2020 / 2021
Exercise Sheet 10

File Input/Output
SOLUTIONS

Content Table

Exercise 1	2
Exercise 2	2
Exercise 3	3
Exercise 4	4
Exercise 5	5
Exercise 6	5
Exercise 7	6
Exercise 8	7
Exercise 9	8
Exercise 10	9
Exercise 11	10
Exercise 12	12

Exercise 1

Write a program that asks the user to introduce numbers, and then saves them in a file named 'numbers.txt'.

Example:

```
Introduce a number: 1
Do you want to introduce more numbers (Y/N)? Y
Introduce a number: 56
Do you want to introduce more numbers (Y/N)? Y
Introduce a number: 34
Do you want to introduce more numbers (Y/N)? Y
Introduce a number: 10
Do you want to introduce more numbers (Y/N)? N
```

Content of the file 'numbers.txt' after execution:

```
1 56 34 10
```

SOLUTION

```
vid = fopen('numbers.txt', 'wt');
if vid == -1
    disp('Error. Could not open the file');
else
    vnumber = input('Introduce a number: ');
    bMore = 'Y';
    while bMore == 'Y'
        fprintf(vid, '%d ', vnumber);
        bMore = input('Do you want to introduce more numbers Y/N)?
', 's');
        if bMore == 'Y'
            vnumber = input('Introduce a number: ');
        end
    end
    fclose(vid);
end
```

Exercise 2

Write a program that asks the user to introduce sentences, and then saves them in a file named 'sentences.txt'. The file should have each sentence written on a different line. Remember that to change lines when using the command *fprintf* we use the control character '\n'

Note: Try to write a program that produces a file that does not have an empty line neither at the end nor at the beginning of the file.

To check the content of the file created just open it with the MATLAB editor.

Example:

```

Introduce a sentence: Windows is an OS
Do you want to introduce more sentences (Y/N)? Y
Introduce a sentence: Mac is another OS
Do you want to introduce more sentences (Y/N)? Y
Introduce a sentence: Linux is an open source OS
Do you want to introduce more sentences (Y/N)? N

```

Content of the file 'sentences.txt' after execution

```

Windows is an OS
Mac is another OS
Linux is an open source OS

```

SOLUTION

```

vid = fopen('sentences.txt', 'wt');
if vid == -1
    disp('Error. Could not open the file');
else
    vsentence = input('Introduce a sentence: ','s');
    % we print the first sentence without \n because we are at the
    % beginning of the file and we don't want to leave a blank line
    fprintf(vid, '%s', vsentence);
    bMore = input('Do you want to introduce more sentences (Y/N)?
    ','s');
    while bMore == 'Y'
        vsentence = input('Introduce a sentence: ','s');
        fprintf(vid, '\n%s', vsentence);
        bMore = input('Do you want to introduce more sentences (Y/N)?
        ','s');
    end
    fclose(vid);
end

```

Exercise 3

Write a program that reads numbers in a file named 'numbers.txt' (the one you created in Exercise 1) and prints the ones which are even on screen. At the end it also prints the total number of even numbers read.

Example:

For a file 'numbers.txt' containing these numbers:

```
1 56 34 10
```

the program prints on screen:

```

56
34
10
There are 3 even numbers in the file

```

SOLUTION

```

fid = fopen('numbers.txt', 'rt');
if fid == -1
    disp('Error. Could not open the file');
else
    vCount = 0;
    while (feof(fid)==0)
        vNum = fscanf(fid, '%d', 1);
        if (rem(vNum, 2)==0)
            fprintf('%d\n', vNum);
            vCount = vCount +1;
        end
    end
    fclose(fid);
    fprintf('There are %d even numbers in the file\n', vCount);
end

```

Exercise 4

Write a program that reads a sequence of numbers in a file named 'numbers.txt', and appends their sum and multiplication at the end of the file.

Example:

Before the process the file contains:

1 56 34 10

After the process the file contains:

1 56 34 10 101 19040

SOLUTION

```

fid = fopen('numbers.txt', 'rt+'); % open the file for reading and
writing
if fid == -1
    disp('Error. Could not open the file');
else
    vSum = 0;
    vMult = 1;
    while (feof(fid)==0)
        vNum = fscanf(fid, '%d', 1);
        vSum = vSum + vNum;
        vMult = vMult * vNum;
    end
    fprintf(fid, '%c', ' '); % add space to separate new data from old
data
    fprintf(fid, '%d', vSum);
    fprintf(fid, '%c', ' '); % add space to separate vSum from vMult
    fprintf(fid, '%d', vMult);
    %fprintf(fid, ' %d %d', vSum, vMult); % alternative to 4 lines above
    fclose(fid);
    disp('File updated correctly');
end

```

Exercise 5

Write a program that reads integer numbers from a file. The program asks the user how many numbers he/she wants to read and prints the maximum number read on screen.

Example:

File content: 1 56 34 84

Execution

```
How many numbers do you want to read? 3
The maximum is 56
```

SOLUTION

```
fid = fopen('numbers.txt', 'rt');
if fid == -1
    disp('Error. Could not open the file');
else
    vMax = -1;
    vLimit = input('How many numbers do you want to read? ');
    for i=1:vLimit
        vNum = fscanf(fid, '%d', 1);
        if vNum > vMax
            vMax = vNum;
        end
    end
    fprintf('The maximum is %d\n', vMax);
    fclose(fid);
end
```

Exercise 6

Modify the previous program so that it takes into account the possibility that the user asks to read more numbers than there are in the file.

Example:

File content: 1 56 34 84

Execution

```
How many numbers do you want to read? 8
There were only 4 numbers in the file
The maximum is 84
```

SOLUTION

```
fid = fopen('numbers.txt', 'rt');
if fid == -1
    disp('Error. Could not open the file');
else
    vCount = 0;
    vMax = -1;
    vLimit = input('How many numbers do you want to read? ');
```

```

% keep reading as long as you are not at the end of the file
% and as long as the number of numbers read is smaller or equal than
% the number the user introduced
while ((feof(fid)==0) && (vCount < vLimit))
    vNum = fscanf(fid, '%d', 1);
    vCount = vCount + 1;
    if vNum > vMax
        vMax = vNum;
    end
end
if vCount < vLimit
    fprintf('There were only %d numbers in the file\n', vCount);
end
fprintf('The maximum is %d\n', vMax);
fclose(fid);
end

```

Exercise 7

Write a function named *searchinfile* that receives as parameters the name of a file and a word, and returns 1 if the word is in the file, 0 if not, and -1 if there was an error when opening the file.

You can test the function with the 'sentence.txt' file you created in Exercise 2. For example, if you run the function with the data of the example in Exercise 2:

```
>> result = searchinfile('sentences.txt','Mac');
the variable result should contain the value 1
```

```
>> var = searchinfile('sentences.txt','iPhone');
the variable result should contain the value 0
```

SOLUTION

```

function [result] = searchinfile (sFileName, sWord)
fid = fopen(sFileName, 'rt');
if fid == -1
    result = -1;
else
    bFound = 0;
    % read as long as you are not at the end of the file
    % and as long as the last word read is not the word you are looking
for
    while ((feof(fid)==0) && (bFound == 0))
        fWord = fscanf(fid, '%s', 1);
        if (strcmp(fWord, sWord) == 1)
            bFound = 1;
        end
    end
    fclose(fid);
    if bFound == 1
        result = 1;
    else
        result = 0;
    end
end

```

```
end
end
```

Exercise 8

Write a program that reads all the words in a file named 'sentences.txt' and prints on screen how many of them start with a vowel and how many with a consonant. Write two versions of the program: one using *fscanf* and another using *textscan*.

Example for the content of the file "sentences.txt" of Exercise 2:

There are 10 words which start with a vowel and 4 which start with a consonant.

SOLUTION using fscanf

```
vfile = fopen('sentences.txt', 'rt');
if vfile == -1
    disp('Error. Could not open the file');
else
    contcons = 0;
    contvowels = 0;
    while (feof(vfile)==0)
        s1 = fscanf(vfile, '%s', 1);
        switch s1(1)
            case {'a','e','i','o','u','A','E','I','O','U'}
                contvowels = contvowels+1;
            otherwise
                contcons = contcons+1;
        end
    end
    fclose(vfile);
    fprintf('There are %d words which start with a vowel and %d which
start with a consonant\n', contvowels, contcons);
end
```

SOLUTION using textscan

```
vfile = fopen('sentences.txt', 'rt');
if vfile == -1
    disp('Error. Could not open the file');
else
    contcons = 0;
    contvowels = 0;
    while (feof(vfile)==0)
        cword = textscan(vfile, '%s', 1);
        s1 = cword{1}{1};
        switch s1(1)
            case {'a','e','i','o','u','A','E','I','O','U'}
                contvowels = contvowels+1;
            otherwise
                contcons = contcons+1;
        end
    end
    fclose(vfile);
```

```
fprintf('There are %d words which start with a vowel and %d which
start with a consonant\n', contvowels, contcons);
end
```

Exercise 9

Write a program that reads a file named 'sentences.txt' and creates a new file 'sentencescopy.txt' that is a copy of the first one but in which all the articles have been removed. Remember that in the English language there are only three articles: *a*, *an*, *the*.

Example of the content of 'sentencescopy.txt' using the example of Exercise 2:

```
Windows is OS
Mac is another OS
Linux is open source OS
```

SOLUTION without line changes

```
vfile = fopen('sentences.txt', 'rt');
vfile2 = fopen('sentencescopy.txt', 'wt');
if (vfile == -1) || (vfile2 == -1)
    disp('Error. Could not open the files.');
```

```
else
    while (feof(vfile)==0)
        cword = textscan(vfile, '%s', 1);
        if ((strcmp(cword{1}{1}, 'a') ~= 1) && (strcmp(cword{1}{1},
'an') ~= 1) && (strcmp(cword{1}{1}, 'the') ~= 1))
            fprintf(vfile2, '%s ', cword{1}{1});
        end
    end
    fclose(vfile);
    fclose(vfile2);
end
```

SOLUTION with line changes

```
vfile = fopen('sentences.txt', 'rt');
vfile2 = fopen('sentencescopy.txt', 'wt');
if (vfile == -1) || (vfile2 == -1)
    disp('Error. Could not open the files.');
```

```
else
    while (feof(vfile)==0)
        line = fgetl(vfile); % read one line without the \n
        % then, split the line into words, and check whether each word
is
        % an article; if not, we write it to vfile2
        while isempty(line) == 0
            [word, rest] = strtok(line);
            if ((strcmp(word, 'a') ~= 1) && (strcmp(word, 'an') ~= 1) &&
(strcmp(word, 'the') ~= 1))
                fprintf(vfile2, '%s ', word);
            end
            line = rest;
        end
    end
end
```



```

        fprintf(vfile2, '\n'); % print newline after finishing
processing the line
    end
    fclose(vfile);
    fclose(vfile2);
end

```

SOLUTION with line changes AND no blank line at the end of the file

```

vfile = fopen('sentences.txt', 'rt');
vfile2 = fopen('sentencescopy.txt', 'wt');
if (vfile == -1) || (vfile2 == -1)
    disp('Error. Could not open the files.');
```

```

else
    line = fgetl(vfile); % read one line without the \n
    % then, split the line into words, and check whether each word is
    % an article; if not, we write it to vfile2
    while isempty(line) == 0
        [word, rest] = strtok(line);
        if ((strcmp(word, 'a') ~= 1) && (strcmp(word, 'an') ~= 1) &&
(strcmp(word, 'the') ~= 1))
            fprintf(vfile2, '%s ', word);
        end
        line = rest;
    end
    while (feof(vfile)==0)
        fprintf(vfile2, '\n'); % print newline
        line = fgetl(vfile);
        while isempty(line) == 0
            [word, rest] = strtok(line);
            if ((strcmp(word, 'a') ~= 1) && (strcmp(word, 'an') ~= 1) &&
(strcmp(word, 'the') ~= 1))
                fprintf(vfile2, '%s ', word);
            end
            line = rest;
        end
    end
    fclose(vfile);
    fclose(vfile2);
end

```

Exercise 10

Write a program that reads from a file named 'race.txt' containing race results in the following format:

Content of file 'race.txt':

```

1111 Duncan Kibet 03-30-28
0234 Heile Gebreselassie 03-35-12
6781 James Kwambai 03-50-01
6331 Juan Pérez 03-55-55

```

The program prints the data on screen as shown below:

<i>Name</i>	<i>Time</i>
<i>D. Kibet</i>	<i>3:30:28</i>
<i>H. Gebreelassie</i>	<i>3:35:12</i>
<i>J. Kwambai</i>	<i>3:50:1</i>
<i>J. Pérez</i>	<i>3:55:55</i>

SOLUTION

```
fid = fopen('race.txt', 'rt');
if (fid == -1)
    disp('Error. Could not open the files.');
```

```
else
    fprintf('Name\tTime\n');
    while not (feof(fid))
        vDorsal = fscanf(fid, '%d', 1);
        vName = fscanf(fid, '%s', 1);
        vSurName = fscanf(fid, '%s', 1);
        vhours = fscanf(fid, '%d', 1);
        vdash1 = fscanf(fid, '%c', 1);
        vmins = fscanf(fid, '%d', 1);
        vdash2 = fscanf(fid, '%c', 1);
        vseconds = fscanf(fid, '%d', 1);
        fprintf('%c. %s\t%d:%d:%d\n', vName(1), vSurName, vhours, vmins,
vseconds);
    end
    fclose(fid);
end
```

ANOTHER SOLUTION

```
fid = fopen('race.txt', 'rt');
if (fid == -1)
    disp('Error. Could not open the files.');
```

```
else
    fprintf('Name\tTime\n');
    while (feof(fid)==0)
        runner.dorsal = fscanf(fid, '%d', 1);
        runner.name = fscanf(fid, '%s', 1);
        runner.surname = fscanf(fid, '%s', 1);
        time = fscanf(fid, '%s', 1);
        runner.hours = str2num(time(1:2));
        runner.mins = str2num(time(4:5));
        runner.seconds = str2num(time(7:8));
        fprintf('%c. %s\t%d:%d:%d\n', runner.name(1), runner.surname,
runner.hours, runner.mins, runner.seconds);
    end
    fclose(fid);
end
```

Exercise 11

Modify Exercise 3 of Exercise sheet 9 so that the output of the program is saved in a file named 'results.txt' in the format depicted below.

Example:

Introduce a name: *Pedro*
 Introduce age: *20*
 Introduce result exam 1: *9*
 Introduce result exam 2: *10*
 Introduce result exam 3: *8*
 Introduce a name: *Ana*
 Introduce age: *18*
 Introduce result exam 1: *7*
 Introduce result exam 2: *8*
 Introduce result exam 3: *9*
 Introduce a name: *Elena*
 Introduce age: *19*
 Introduce result exam 1: *10*
 Introduce result exam 2: *10*
 Introduce result exam 3: *9*
 Introduce a name:
 You've finished introducing names.

Introduce a name: *Pedro*
 Information about Pedro's results is now saved

Introduce a name: *Ana*
 Information about Ana's results is now saved

Introduce a name:
 Ok, bye!

Content of the file 'results.txt' after execution

Pedro * 9.00 10.00 8.00 * 9.00 * 0.11
 Ana * 7.00 8.00 9.00 * 8.00 * -0.89

SOLUTION

```

varName = input('Introduce a name: ', 's');
cont = 0;
vClassMeans = [];
while isempty(varName) == 0
    varAge = input('Introduce an age: ');
    varresults = zeros(1,3);
    for i=1:3
        fprintf('Introduce result exam %d: ', i);
        varresults(i) = input('');
    end
    cont = cont + 1;
    students(cont).name = varName;
    students(cont).age = varAge;
    students(cont).results = varresults;
    students(cont).mean = sum(varresults)/3;
    vClassMeans(cont) = students(cont).mean;
    varName = input('Introduce a name: ', 's');
end
classMean = sum(vClassMeans)/cont;
disp('You've finished introducing names');
vid = fopen('results.txt', 'wt');
  
```

```

if (vid == -1)
    disp('Error when opening the file to save the information');
else
    bFirstLine = 1;
    varName = input('Introduce a name: ', 's');
    while (isempty(varName) == 0)
        posStud = findStudent(students, varName);
        if (posStud == 0)
            disp('Sorry, nobody with that name');
        else
            if bFirstLine == 1
                % print the first line without /n to not create a
                % blank line at the beginning of the file
                fprintf(vid, '%s * ', varName);
                bFirstLine = 0;
            else
                % for the rest of the lines: write /n to change lines
                fprintf(vid, '\n%s * ', varName);
            end
            for j=1:3
                fprintf(vid, '%.2f ', students(posStud).results(j));
            end
            fprintf(vid, '* %.2f ', students(posStud).mean);
            fprintf(vid, '* %.2f ', students(posStud).mean-classMean);
            fprintf('Information about %s's results is now saved\n',
varName);
        end
        varName = input('Introduce a name: ', 's');
    end
    fclose(vid);
    disp('Ok, bye!');
end

function [pos] = findStudent (ststudents, sName)
% function that receives a vector of structures 'students' and a name
% and returns the position of the vector in which a student with that
% name is found
% if no student is found with that name the function returns 0
cont = 1;
bFound = 0;
pos = 0;
while ((cont <= length(ststudents)) && (bFound == 0))
    if (strcmp(ststudents(cont).name, sName) == 1)
        pos = cont;
        bFound = 1;
    else
        cont = cont + 1;
    end
end
end

```

Exercise 12

Write a program that asks the user to introduce a number between 1 and 3 and prints the name of the student who has the highest result on the corresponding exam on screen. The program reads the information from the files 'results.txt' created in the previous exercise (Exercise 11).

- a) Solve the problem considering that there are no students with the same mark on the same exam.

Example.

Content of the file 'results.txt':

```
Pedro * 9.00 10.00 8.00 * 9.00 * 0.11
Ana * 7.00 8.00 9.00 * 8.00 * -0.89
Elena * 10.00 10.00 9.00 * 9.67 * 0.78
```

Example of execution:

Introduce the exam number: 1

The student with the highest mark on exam 1 is: Elena

- b) Solve the problem considering that it is possible that two or more students had the same mark on the same exam.

This means that when two or more students obtained the highest mark on the exam selected by the user, the program should print all of their names, one after the other.

Example.

Content of the file 'results.txt':

```
Pedro * 9.00 10.00 8.00 * 9.00 * 0.11
Ana * 7.00 8.00 9.00 * 8.00 * -0.89
Elena * 10.00 10.00 9.00 * 9.67 * 0.78
```

Example of execution:

Introduce the exam number: 2

The students with the highest marks on exam 2 are: Pedro – Elena

SOLUTION A

```
vExam = input('Introduce the exam number: ');
vid = fopen('results.txt', 'rt');
if (vid == -1)
    disp('Error when opening the file');
else
    greaterstf = -1;
    nameBest = '';
    while (feof(vid) == 0)
        student.name = fscanf(vid, '%s', 1);
        vstar = fscanf(vid, '%s', 1);
        student.exams = zeros(1,3); % create a vector of 0s to store the
exam results of the corresponding student
        student.exams(1) = fscanf(vid, '%f', 1);
        student.exams(2) = fscanf(vid, '%f', 1);
        student.exams(3) = fscanf(vid, '%f', 1);
        vrest = fgets(vid); % not interested in the rest of the
information, but needs to be read to move pointer forward
        if student.exams(vExam) > greaterstf
            greaterstf = student.exams(vExam);
```

```
        nameBest = student.name;
    end
end
fprintf('The student with the highest mark for exam %d is: %s\n',
vExam, nameBest);
fclose(vid);
end
```

SOLUTION B

```
vExam = input('Introduce the exam number: ');
vid = fopen('results.txt', 'rt');
if (vid == -1)
    disp('Error when opening the file');
else
    greaterstf = -1;
    nameBest = '';
    while (feof(vid) == 0)
        student.name = fscanf(vid, '%s', 1);
        vstar = fscanf(vid, '%s', 1);
        student.exams = zeros(1,3); % create a vector of 0s to store the
exam results of the corresponding student
        student.exams(1) = fscanf(vid, '%f', 1);
        student.exams(2) = fscanf(vid, '%f', 1);
        student.exams(3) = fscanf(vid, '%f', 1);
        vrest = fgets(vid); % not interested in the rest of the
information, but needs to be read to move pointer forward
        if student.exams(vExam) > greaterstf
            greaterstf = student.exams(vExam);
            nameBest = student.name;
        elseif student.exams(vExam) == greaterstf
            nameBest = [nameBest '-' student.name];
        end
    end
    fprintf('The student with the highest mark for exam %d is: %s\n',
vExam, nameBest);
fclose(vid);
end
```