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

Review Exercises III
– SOLUTIONS –

Content Table

Exercise 1

Write a program that reads numbers from 1 to 9 in a file named “numbers.txt”, counts how many times each of them appears, and creates a file named “counting.txt” with the result.

Example:

Content of the file *numbers.txt*

```
2 3 4 2 3 4 2 2 7 2 3 5 3 2 3 4 5 6
```

Content of the file *counting.txt* after execution

```
1 – 0 times
2 – 6 times
3 – 5 times
4 – 3 times
5 – 2 times
6 – 1 times
7 – 1 times
8 – 0 times
9 – 0 times
```

SOLUTION

```
fid = fopen('numbers.txt','rt');
count = zeros(1,9);
vRead = [];
% First we read all the information in the file
while ~feof(fid)
    vnumber = fscanf(fid,'%d', 1);
    vRead = [vRead vnumber];
end
fclose(fid);
% Then we count how many times the numbers 1 to 9 appear in the vector
for i=1:9
    for j=1:length(vRead)
        if i == vRead(j)
            count(i) = count(i) + 1;
        end
    end
end
% Finally we write the result in the output file
fid2 = fopen('counting.txt','wt');
for i=1:8
    fprintf(fid2, '%d - %d times\n',i, count(i));
end
fprintf(fid2, '%d - %d times', 9, count(9));
fclose(fid2);
```

Exercise 2

Write a program for copying information introduced by the user into a text file. After program execution, all the information introduced by the user should be stored in a text file called ‘output.txt’. When the user introduces a line containing only the word *bye*, the program stops. In case the program is executed again, the new information should be appended to the file.

Example of information introduced by the user:

```
This is a test
for exercise 9
cats text cats
bye
```

Content of the file output.txt after execution:

This is a test
for exercise 9
cats text cats

SOLUTION

```
vfile = fopen('output.txt', 'at+');
if vfile == -1
    disp('Error. Could not open the files.');
else
    sline = input('', 's');
    bFirst = 1;
    while not(strcmp(sline, 'bye') == 1)
        if bFirst == 1
            fprintf(vfile, '%s', sline);
            bFirst = 0;
        else
            fprintf(vfile, '\n%s', sline);
        end
        sline = input('','s');
    end
    fclose(vfile);
end
```

Exercise 3

Write a program that reads an input text file and produces an exact copy of it only using uppercase characters.

SOLUTION

```
vfile = fopen('textfile1.txt', 'rt');
vfile2 = fopen('textfile2.txt', 'wt');
bFirst = 1;
while not(feof(vfile))
    vline= fgetl(vfile);
    if bFirst == 1
        fprintf(vfile2, '%s', upper(vline));
        bFirst = 0;
    else
        fprintf(vfile2, '\n%s', upper(vline));
    end
end
fclose(vfile);
fclose(vfile2);
```

Exercise 4

Write a program that asks the user to introduce the names of two text files, compares their content and indicates if they are equal.

SOLUTION

```
vname1 = input('Introduce the name of the first file you want to compare:
','s');
vname2 = input('Introduce the name of the second file you want to
compare: ','s');
vfile1  = fopen(vname1, 'rt');
vfile2  = fopen(vname2, 'rt');
if vfile1 == -1 || vfile2 == -1
    disp('File not found');
else
    bEqual = 1;
    while (feof(vfile1)==0) && (feof(vfile2)==0) && (bEqual == 1)
        sline1 = fgets(vfile1);
        sline2 = fgets(vfile2);
        if (strcmp(sline1,sline2) == 0)
            bEqual = 0;
        end
    end
    % Now we need to know if the while finished because they are equal
    % or because we reached the end of one of the files
    % and the other file includes more information
    if (bEqual == 1)
        if ((feof(vfile1)==1) && (feof(vfile2)==0)) || (feof(vfile1)==0
&& (feof(vfile2)==1))
            bEqual = 0;
        end
    end
    if bEqual == 1
        disp('The two files have the same content');
    else
        disp('The files are different');
    end
    fclose(vfile1);
    fclose(vfile2);
end
```

Exercise 5

Write a program for writing a text file with the times of the runners of a race. The information about the runners is stored in a file named ‘hobbitrun.txt’ following the format:

runnerid name surname age gender

where *runnerid* and *age* are integers, *name* and *surname* are strings, and *gender* is either an F (female) or an M (male).
The times of the runners are stored in another file named ‘times.txt’ following the format:

runnerid hours:minutes:seconds

There are three categories of runners according to their age:

- **Sub-23**, for runners below 23 years old
- **Elite**, for runners between 23 and 60 years old
- **Masters**, for runners older than 60 years old

The program should organize the information in the files and produce six files:

- m-sub23.txt: times of the male runners of the category sub-23
- *f-sub23.txt*: times of the female runners of the category sub-23
- *m-elite.txt*: times of the male runners of the category elite
- *f-elite.txt*: times of the female runners of the category elite
- *m-masters.txt*: times of the male runners of the category masters
- *f-masters.txt*: rimes of the female runners of the category master

Each file contains the information about the runners of the corresponding category, including their times.

FUNCTION

```
function [indexrdo, err] = findrunner(vrunners, ndorsal)
err = 'OK';
pos = 1;
indexrdo = -1;
while (pos <= length(vrunners)) && (indexrdo == -1)
    if vrunners(pos).dorsal == ndorsal
        indexrdo = pos;
    else
        pos = pos +1;
    end
end
if indexrdo == -1
    err = 'NOTFOUND';
end
end
```

MAIN PROGRAM

```
clear vstimes;
clear runners;
clear C;
fid = fopen('times.txt','rt');
fid2 = fopen('hobbitrun.txt','rt');
if (fid == -1) || (fid2 == -1)
    disp('Error. Could not open the files.');
else
    % Read the times
    cont = 0;
    while feof(fid) ==0
        vDorsal = fscanf(fid,'%d',1);
        vhours = fscanf(fid,'%d',1);
        vdash = fscanf(fid,'%c',1);
        vmins = fscanf(fid,'%d',1);
        vdash = fscanf(fid,'%c',1);
        vseconds = fscanf(fid,'%d',1);
        cont = cont + 1;
        vstimes(cont).dorsal = vDorsal;
        vstimes(cont).hours = vhours;
        vstimes(cont).mins  = vmins;
        vstimes(cont).seconds = vseconds;
        vstimes(cont).total = vhours*3600 + vmins*60 + vseconds;
    end
    % Read the data of the runners
    cont = 0;
    while feof(fid2)==0
        C = textscan(fid2,'%d %s %s %d %c',1);
        cont = cont +1;
        runners(cont).dorsal = C{1}(1);
        runners(cont).name = C{2}{1};
        runners(cont).surname = C{3}{1};
        runners(cont).age = C{4}(1);
        runners(cont).gender = C{5}(1);
    end
    % Create the rankings, read from the times vector of structures
    % Search for corresponding runner in runners structure
    fra1 = fopen('m-sub23.txt','wt');
    bFirst1 = 1;
    fra2 = fopen('f-sub23.txt','wt');
    bFirst2 = 1;
    fra3 = fopen('m-elite.txt','wt');
    bFirst3 = 1;
    fra4 = fopen('f-elite.txt','wt');
    bFirst4 = 1;
    fra5 = fopen('m-masters.txt','wt');
    bFirst5 = 1;
    fra6 = fopen('f-masters.txt','wt');
    bFirst6 = 1;
    for i=1:length(vstimes)
        [indexrun, err] = findrunner(runners, vstimes(i).dorsal);
        if strcmp(err, 'OK') == 1
            % Store in variable fra the identifier of the file that
corresponds to the gender and age of the runner
            if runners(indexrun).age < 23
                if runners(indexrun).gender == 'M'
                    fra = fra1;
                    if bFirst1 == 1
                        bFirst = 1;
                        bFirst1 = 0;
                    end
                else
                    fra = fra2;
                    if bFirst2 == 1
                        bFirst = 1;
                        bFirst2 = 0;
                    end
                end
            elseif runners(indexrun).age < 60
                if runners(indexrun).gender == 'M'
                    fra = fra3;
                    if bFirst3 == 1
                        bFirst = 1;
                        bFirst3 = 0;
                    end
                else
                    fra = fra4;
                    if bFirst4 == 1
                        bFirst = 1;
                        bFirst4 = 0;
                    end
                end
            elseif runners(indexrun).gender == 'M'
                fra = fra5;
                if bFirst5 == 1
                    bFirst = 1;
                    bFirst5 = 0;
                end
            else
                fra = fra6;
                if bFirst6 == 1
                    bFirst = 1;
                    bFirst6 = 0;
                end
            end
        end
        if bFirst == 1
            fprintf(fra,'%d %s %s %d-%d-%d', runners(indexrun).dorsal,
runners(indexrun).name, runners(indexrun).surname, vstimes(i).hours,
vstimes(i).mins, vstimes(i).seconds);
            bFirst = 0;
        else
            fprintf(fra,'\n%d %s %s %d-%d-%d', runners(indexrun).dorsal,
runners(indexrun).name, runners(indexrun).surname, vstimes(i).hours,
vstimes(i).mins, vstimes(i).seconds);
        end
    end
end
fclose(fra1);
fclose(fra2);
fclose(fra3);
fclose(fra4);
fclose(fra5);
fclose(fra6);
fclose(fid);
fclose(fid2);
```

Exercise 6

IKEA asked you to write a program to help their customers when selecting the furniture that best suits their necessities.

The IKEA catalog is stored in file ‘IKEA.txt’ following the format:

productID name type weight height price color

where *productID* is an integer, *name*, *type* and *color* are strings and *weight*, *height* and *price* are numbers with decimals.

The program should allow the user to:
1.- Display the information about a particular product identified by its name on screen. The program asks the user to introduce the name of the product and displays the product’s information.
2.- Display all the products of a specific type. The user should be able to filter the search by weight and height (display only the products whose dimensions are smaller than a given weight and height introduced by the user).

Note: Try to solve the problem splitting the code into functions. For example: searchByProduct, searchByCategory, …

Example of execution:

```
Do you want to display (A) a product (B) all the products of a
category or (Q) Quit(Q)? A
Introduce the name of the product: GRONKULLA
Product information:
 Name: GRONKULLA
 Type: Table
 Weigth: 1.50
 Height: 2.00
 Price: 250.00
 Color: red

Do you want to display (A) a product (B) all the products of a
category or (Q) Quit(Q)? B
Introduce a category: Curtain
Introduce a maximum weight: 4
Introduce a maximum height: 6
Product list
 Name: RIKTIG
 Type: Curtain
 Weigth: 3.00
 Height: 3.00
 Price: 50.00
 Color: blue

Do you want to display (A) a product (B) all the products of a
category or (Q) Quit(Q)? Q
Bye!
```

FUNCTIONS

```
function [pos] = searchByProduct (vproducts, sname)
pos = 0;
bFound = 0;
while ((bFound == 0) && (pos < length(vproducts)))
    pos = pos +1;
    if (strcmp(vproducts(pos).name, sname) == 1)
        bFound = 1;
    end
end
if bFound == 0
    pos = -1; % If not found, return -1
end
end


function [vect] = searchByCategory (vproducts, scat, maxw, maxh)
vect = [];
for i=1:length(vproducts)
    if ((strcmp(vproducts(i).category, scat) == 1) &&
(vproducts(i).weight <= maxw) && (vproducts(i).height <= maxh))
        vect = [vect i];
    end
end
end


function [] = displayProductInfo (vproducts, pos)
fprintf('\n Name: %s',vproducts(pos).name);
fprintf('\n Type: %s',vproducts(pos).category);
fprintf('\n Weight: %.2f',vproducts(pos).weight);
fprintf('\n Height: %.2f',vproducts(pos).height);
fprintf('\n Price: %.2f',vproducts(pos).price);
fprintf('\n Color: %s',vproducts(pos).color);
end
```

MAIN PROGRAM

```
clear;
fid = fopen('IKEA.txt','rt');
if fid == -1
    disp('Error when opening the file');
else
    i = 0;
    while feof(fid)==0
        C = textscan(fid,'%s %s %s %f %f %f %s',1);
        i = i +1;
        product(i).id = C{1}{1};
        product(i).name = C{2}{1};
        product(i).category = C{3}{1};
        product(i).weight = C{4}(1);
        product(i).height = C{5}(1);
        product(i).price = C{6}(1);
        product(i).color = C{7}{1};
    end
    opt = 'x'; % Can be anything, necessary to get into while loop
    while opt ~= 'Q'
```

```
        fprintf('\n');
        opt = input('Do you want to display (A) a product (B) all the
products of a category or (Q) Quit? ','s');
        while ((opt ~= 'A') && (opt ~= 'B') && (opt ~= 'Q'))
            disp('Wrong option');
            opt = input('Do you want to display (A) a product (B) all the
products of a category or (Q) Quit? ','s');
        end
        if opt == 'A'
            name = input('Introduce the name of the product: ','s');
            index = searchByProduct(product,name);
            if index == -1
                disp('Error, product not found');
            else
                disp ('Product information:');
                displayProductInfo(product,index);
            end
        elseif opt == 'B'
            category = input('Introduce a category: ','s');
            maxw = input('Introduce a maximum weight: ');
            maxh = input('Introduce a maximum height: ');
            vindex = searchByCategory(product,category,maxw,maxh);
            disp('Product list');
            for j=1:length(vindex)
                displayProductInfo(product,vindex(j));
            end
        end
    end
end
```

Exercise 7

Write a program that reads numbers in a file named “numbers2.txt”, counts how many times each of them appears, and creates a file named “counting2.txt” with the result. Note: it will only print the numbers which appear in the file (this is a different problem as the one presented in the first exercise).

Example:

Content of the file numbers2.txt

```
29 3 20 3 4 23 2 7 2 23 5 3 2 31 4 5 60
```

Content of the file counting2.txt after the execution

```
29 - 1 times
3 - 3 times
20 - 1 times
4 - 2 times
23 - 2 times
2 - 3 times
7 - 1 times
5 - 2 times
31 - 1 times
60 - 1 times
```

SOLUTION

```
fid = fopen('numbers2.txt','rt');
% strcount is a structure in which we will store the numbers and the
times they appear
clear strcount;
vRead = [];
% first, we read all the information in the file
while feof(fid) ==0
    vnumber = fscanf(fid,'%d',1);
    vRead = [vRead vnumber];
end
fclose(fid);
% index indicates the position of the vector of structures in which we
are putting the information
index = 0;
while length(vRead) >= 1
    index = index + 1;
    % we take the first number from vRead and we set the count to one
    strcount(index).number = vRead(1);
    strcount(index).times = 1;
    % then, we delete the number from the vector
    vRead(1) = [];
    % and count how many times the number appears in the rest of the
vector
    j = 1;
    while j <= length(vRead)
        if strcount(index).number == vRead(j)
            % each time we find the number in the vector we increase its
counter and delete it from the vector
            strcount(index).times = strcount(index).times + 1;
            vRead(j) = [];
        else
            j = j + 1;
        end
    end
end

% finally, we write the result to the output file
fid2 = fopen('counting2.txt','wt');
for i=1:length(strcount)-1
    fprintf(fid2, '%d - %d times\n',strcount(i).number,
strcount(i).times);
end
% we make sure that the last line doesn't create an extra blank line
fprintf(fid2, '%d - %d times',strcount(length(strcount)).number,
strcount(length(strcount)).times);
fclose(fid2);
```

Exercise 8

Write a program that reads from a file named *courses.txt* containing a list of different courses for different degrees, as well as whether the course will be taught in English, like this:

ID	COURSE	TEACHER	DEGREE	ENGLISH	CREDITS
1002356	Programming	Mendoza	Engineering	N	3
1004562	Poetry	Diaz	English	Y	5
1004578	Postmodernism	Perez	English	Y	6
1007658	Statistics	Montoya	Mathematics	Y	8
1005123	Optics	Mojica	Physics	Y	5
1006235	Classics	Aguilar	English	N	4
1002568	Electronics	Velasquez	Engineering	Y	4
1005687	Biophysics	Tormo	Physics	N	4
1004854	Modernism	Sanchez	English	Y	4

The program asks the user to introduce from which degree they would like to choose a course, and whether he/she would like the course to be in English. Next, the program will display the available courses according to the user's criteria. The user is then asked to choose from the available courses according to their criteria until they introduce an empty string.
At the end, the program generates the file *choices.txt* that lists the courses chosen, as well as the total number of credits.

Example of execution:

```
From which degree would you like to choose a course? English
Would you like the course to be in English (Y/N)? Y
These are the courses available according to your criteria:
English: Poetry, Credits: 5
English: Postmodernism, Credits: 6
English: Modernism, Credits: 4
Introduce name of course: Poetry
Introduce name of course: Modernism
Introduce name of course:
File created successfully!
```

Content of *choices.txt* after execution:

ID	COURSE	TEACHER	DEGREE	ENGLISH	CREDITS
1004562	Poetry	Diaz	English	Y	5
1004578	Modernism	Perez	English	Y	4
TOTAL CREDITS: 9					

SOLUTION

```
clear;
clc;
fid = fopen('courses.txt', 'rt');
if fid < 0
    disp('Error opening file');
else
    %READ FILE AND STORE CONTENTS
    i = 0;
    firstline = fgets(fid);
    while (feof(fid)==0)
        i = i + 1;
        info = textscan(fid, '%s %s %s %s %s %d', 1);
        courses(i).id =  info{1}{1};
        courses(i).course = info{2}{1};
        courses(i).teacher = info{3}{1};
        courses(i).degree = info{4}{1};
        courses(i).english = info{5}{1};
        courses(i).credits = info{6}(1);
    end
end
fclose(fid);

%DISPLAY AVAILABLE COURSES
disp('The courses available are');
for i = 1:length(courses)
    fprintf('%s: %s (English: %s, Credits: %d)\n', courses(i).degree,
courses(i).course, courses(i).english, courses(i).credits);
end

%ASK FOR DEGREE AND ENGLISH
degree = input('From which degree would you like to choose a course? ',
's');
english = input('Would you like the course to be in English (Y/N)? ',
's');

%LIST AVAILABLE COURSES
disp('These are the courses available according to your criteria:');
for i = 1:length(courses)
    if strcmpi(courses(i).degree, degree) && courses(i).english ==
english
        fprintf('%s: %s, Credits: %d\n', courses(i).degree,
courses(i).course, courses(i).credits);
    end
end

%CHOOSE COURSE
course = input('Introduce name of course: ', 's');
choices = [];
credits = 0;
while (isempty(course) == 0)
    for i=1:length(courses)
        if strcmpi(courses(i).course, course) &&
strcmpi(courses(i).degree, degree) && courses(i).english == english
            choices = [choices courses(i)];
            credits = credits + courses(i).credits;
        end
    end
    course = input('Introduce name of course: ', 's');
end

%GENERATE OUTPUT
fid2 = fopen('choices.txt', 'wt');
fprintf(fid2, firstline);
for i = 1:length(choices)
    fprintf(fid2, '%s %s %s %s %s %d\n', choices(i).id,
choices(i).course, choices(i).teacher, choices(i).degree,
choices(i).english, choices(i).credits);
end
fprintf(fid2, 'TOTAL CREDITS: %d', credits);
fclose(fid2);
disp('File created successfullly!');
```

Exercise 9

Modify the previous program to control the possibility that the user selects the same course more than once. In this case the program should display an error message and ignore the last selection.

Example of execution:

```
From which degree would you like to choose a course? English
Would you like the course to be in English (Y/N)? Y
These are the courses available according to your criteria:
English: Poetry, Credits: 5
English: Postmodernism, Credits: 6
English: Modernism, Credits: 4
Introduce name of course: Poetry
Introduce name of course: Modernism
Introduce name of course: Poetry
Sorry, you already selected the course Poetry of the degree
English.
Introduce name of course:
File created successfully!
```

Content of *choices.txt* after execution:

```
ID          COURSE     TEACHER  DEGREE     ENGLISH     CREDITS
1004562     Poetry     Diaz     English    Y           5
1004578     Modernism  Perez    English    Y           4
TOTAL CREDITS: 9
```

SOLUTION

```
clear;
clc;
fid = fopen('courses.txt', 'rt');
if fid < 0
    disp('Error opening file');
else
    %READ FILE AND STORE CONTENTS
    i = 0;
    firstline = fgets(fid);
    while (feof(fid) ==0)
        i = i + 1;
        info = textscan(fid, '%s %s %s %s %s %d', 1);
        courses(i).id = info{1}{1};
        courses(i).course = info{2}{1};
        courses(i).teacher = info{3}{1};
        courses(i).degree = info{4}{1};
        courses(i).english = info{5}{1};
        courses(i).credits = info{6}(1);
    end
end
fclose(fid);

%DISPLAY AVAILABLE COURSES
disp('The courses available are');
for i = 1:length(courses)
    fprintf('%s: %s (English: %s, Credits: %d)\n', courses(i).degree,
courses(i).course, courses(i).english, courses(i).credits);
end

%ASK FOR DEGREE AND ENGLISH
```

```
degree = input('From which degree would you like to choose a course? ',
's');
english = input('Would you like the course to be in English (Y/N)? ',
's');

%LIST AVAILABLE COURSES
disp('These are the courses available according to your criteria:');
for i = 1:length(courses)
    if strcmpi(courses(i).degree, degree) && courses(i).english ==
english
        fprintf('%s: %s, Credits: %d\n', courses(i).degree,
courses(i).course, courses(i).credits);
end
end

%CHOOSE COURSE
course = input('Introduce name of course: ', 's');
choices = [];
credits = 0;
while (isempty(course) == 0)
    for i=1:length(courses)
        if strcmpi(courses(i).course, course) &&
strcmpi(courses(i).degree, degree) && courses(i).english == english
            % Check that the course has not been selected already
            bFound = 0;
            j = 0;
            while (bFound == 0) && (j < length(choices))
                j = j + 1;
                if strcmpi(choices(j).course, course)
                    bFound = 1;
                end
            end
            if bFound == 1
                fprintf('Sorry, you already selected the course %s of the
degree %s.\n', choices(j).course, choices(j).degree);
            else
                choices = [choices courses(i)];
                credits = credits + courses(i).credits;
            end
        end
    end
    course = input('Introduce name of course: ', 's');
end

%GENERATE OUTPUT
fid2 = fopen('choices.txt', 'wt');
fprintf(fid2, firstline);
for i = 1:length(choices)
    fprintf(fid2, '%s %s %s %s %s %d\n', choices(i).id,
choices(i).course, choices(i).teacher, choices(i).degree,
choices(i).english, choices(i).credits);
end
fprintf(fid2, 'TOTAL CREDITS: %d', credits);
fclose(fid2);
disp('File created successfullly!');
```

Exercise 10

Write a program that creates a file containing a list of offers for Black Friday. The program processes a list of products contained in a file named *products.txt* following this format:

REF	PRODUCT	BRAND	CATEGORY	PRICE
1112	Mouse	XC	IT	20.00
6678	TV	Samsang	Image	700.20
6789	Computer	BlueRang	IT	3500.00
2345	Speaker	Raft	Sound	430.50
6782	VRHeadset	Raft	IT	600.10
7763	Controller	Namark	Sound	234.20
7363	Tablet	BlueRang	IT	434.00
1234	Computer	Namark	IT	1199.90
4522	Monitor	Samsang	Image	789.00

and a list of discounts to apply contained in a file named *discounts.txt* like this:

TYPE	NAME	DISCOUNT
Product	Mouse	1.00
Category	IT	5.00
Brand	Samsang	10.00
Category	Sound	2.00

When the discount type is *Category* the discount should be applied to all the products of the specified category, when the type is *Brand* to all the products of that Brand, if it’s *Produc*t to that specific product. The discounts should be applied in the order that they appear in the file, and the price of a product might be reduced as a result of the application of more than one discount.

The program should generate a file *offers.txt* with a similar format as the file *products.txt* but containing only the products whose prices have been reduced by discounts, and their current price (after applying the discounts).

For example, the file *offers.txt* for the examples of files *products.txt* and *discounts.txt* above should be:

REF	PRODUCT	BRAND	CATEGORY	PRICE
1112	Mouse	XC	IT	18.81
6789	Computer	BlueRang	IT	3325.00
2345	Speaker	Raft	Sound	421.89
6782	VRHeadset	Raft	IT	570.10
7763	Controller	Namark	Sound	229.52
7363	Tablet	BlueRang	IT	412.30
1234	Computer	Namark	IT	1139.91
4522	Monitor	Samsang	Image	710.10

SOLUTION

```
clear;
clc;
fid = fopen('products.txt', 'rt');
if fid < 0
    disp('Error opening file');
else
```

```
    i = 0;
    firstline = fgets(fid);
    while (feof(fid)==0)
        i = i + 1;
        info = textscan(fid, '%d %s %s %s %.2f', 1);
        products(i).id = info{1}(1);
        products(i).name = info{2}{1};
        products(i).brand = info{3}{1};
        products(i).category = info{4}{1};
        products(i).price = info{5}(1);
        products(i).newprice = info{5}(1);
    end
end
fclose(fid);
disp('Available products:');
for i=1:length(products)
    fprintf('%d %s %s %s %.2f\n', products(i).id, products(i).name,
products(i).brand, products(i).category, products(i).price');
end

fid2 = fopen('discounts.txt', 'rt');
if fid2 < 0
    disp('Error opening file');
else
    i = 0;
    firstline = fgets(fid2);
    while (feof(fid2)==0)
        info = textscan(fid2, '%s %s %.2f', 1);
        discount.type = info{1}{1};
        discount.name = info{2}{1};
        discount.value = info{3}(1);
        for j=1:length(products)
            if ((strcmp(discount.type, 'Product') == 1) &&
(strcmp(products(j).name, discount.name) == 1))
                products(j).newprice = products(j).newprice -
(discount.value*products(j).newprice/100);
            elseif ((strcmp(discount.type, 'Category') == 1) &&
(strcmp(products(j).category, discount.name) == 1))
                products(j).newprice = products(j).newprice -
(discount.value*products(j).newprice/100);
            elseif ((strcmp(discount.type, 'Brand') == 1) &&
(strcmp(products(j).brand, discount.name) == 1))
                products(j).newprice = products(j).newprice -
(discount.value*products(j).newprice/100);
            end
        end
    end
end
fclose(fid2);
fid3 = fopen('offers.txt', 'wt');
if fid3 < 0
    disp('Error opening file');
else
    bFirst = 1;
    for i=1:length(products)
        if (products(i).newprice ~= products(i).price)
            if (bFirst == 1)
                fprintf(fid3,'%d %s %s %s %.2f', products(i).id,
products(i).name, products(i).brand, products(i).category,
products(i).newprice);
                bFirst = 0;
            else
```

```
                fprintf(fid3,'\n%d %s %s %s %.2f',products(i).id, 
products(i).name, products(i).brand, products(i).category, 
products(i).newprice);
            end
        end
    end
end
fclose(fid3);
```

Exercise 11

Modify the previous program so that it allows the user to select which discounts to apply. The program offers three types of discounts: by category, by brand and by product. The user selects one of them, and then the program asks again if he/she wants to apply another discount until the user introduces an empty string. Then, the program generates a file *offers.txt* considering only the discounts in the *discounts.txt* file that match the user selection.

Example of execution:

```
** Types of discounts available **
- Product
- Category
- Brand
Introduce which type of discount you want to apply? Product
Introduce which type of discount you want to apply? Brand
Introduce which type of discount you want to apply?
```

Content of the file *offers.txt* after running the program:

```
REF         PRODUCT     BRAND CATEGORY        PRICE
1112        Mouse       XC          IT        19.80
4522        Monitor     Samsang Image         710.10
```

SOLUTION

```
clear;
clc;
fid = fopen('products.txt', 'rt');
if fid < 0
    disp('Error opening file');
else
    i = 0;
    firstline = fgets(fid);
    while (feof(fid) ==0)
        i = i + 1;
        info = textscan(fid, '%d %s %s %s %.2f', 1);
        products(i).id = info{1}(1);
        products(i).name = info{2}{1};
        products(i).brand = info{3}{1};
        products(i).category = info{4}{1};
        products(i).price = info{5}(1);
        products(i).newprice = info{5}(1);
    end
end
fclose(fid);
disp('Available products:');
```

```
for i=1:length(products)
    fprintf('%d %s %s %s %.2f\n', products(i).id, products(i).name,
products(i).brand, products(i).category, products(i).price');
end

disp('** Types of discounts available **');
disp('- Product');
disp('- Category');
disp('- Brand');
sString = input('Introduce which type of discount you want to apply:
','s');
count = 0;
while isempty(sString) == 0
    count = count + 1;
    selection{count} = sString;
    sString = input('Introduce which type of discount you want to apply:
','s');
end

fid2 = fopen('discounts.txt', 'rt');
if fid2 < 0
    disp('Error opening file');
else
    i = 0;
    firstline = fgets(fid2);
    while (feof(fid2) == 0)
        info = textscan(fid2, '%s %s %.2f', 1);
        discount.type = info{1}{1};
        discount.name = info{2}{1};
        discount.value = info{3}(1);
        bFound = 0;
        l = 0;
        while (bFound == 0) && (l < length(selection))
            l = l + 1;
            if strcmpi(selection{l}, discount.type)
                bFound = 1;
            end
        end
        if bFound == 1
            for j=1:length(products)
                if ((strcmp(discount.type, 'Product') == 1) &&
(strcmp(products(j).name, discount.name) == 1))
                    products(j).newprice = products(j).newprice -
(discount.value*products(j).newprice/100);
                elseif ((strcmp(discount.type, 'Category') == 1) &&
(strcmp(products(j).category, discount.name) == 1))
                    products(j).newprice = products(j).newprice -
(discount.value*products(j).newprice/100);
                elseif ((strcmp(discount.type, 'Brand') == 1) &&
(strcmp(products(j).brand, discount.name) == 1))
                    products(j).newprice = products(j).newprice -
(discount.value*products(j).newprice/100);
                end
            end
        end
    end
end
fclose(fid2);

fid3 = fopen('offers.txt', 'wt');
if fid3 < 0
    disp('Error opening file');
```

```
else
    bFirst = 1;
    for i=1:length(products)
        if (products(i).newprice ~= products(i).price)
            if (bFirst == 1)
                fprintf(fid3,'%d %s %s %s %.2f',products(i).id, 
products(i).name, products(i).brand, products(i).category, 
products(i).newprice);
                bFirst = 0;
            else
                fprintf(fid3,'\n%d %s %s %s %.2f',products(i).id, 
products(i).name, products(i).brand, products(i).category, 
products(i).newprice);
            end
        end
    end
end
fclose(fid3);
```

Exercise 12

Write a program that reads from a file named *schools.txt* that lists different dance style courses for two different dance schools, *CityDance* and *DanceCenter,* like this:

STYLE	LEVEL	TEACHER	SCHOOL	PRICE
hiphop	beginner	Perez	CityDance	70
salsa	advanced	Diaz	CityDance	80
contemporary	beginner	Aguilar	DanceCenter	70
breakdance	beginner	Montoya	CityDance	75
flamenco	intermediate	Sanchez	DanceCenter	85
hiphop	intermediate	Benavides	DanceCenter	80
ballet	junior	Mojica	CityDance	75
contemporary	intermediate	Tormo	CityDance	85
tapdance	beginner	Montero	DanceCenter	60
hiphop	beginner	Benavides	DanceCenter	75
salsa	advanced	Diaz	CityDance	75
salsa	intermediate	Garcia	DanceCenter	85

The program will generate a file *schoolcourses.txt* containing a list of the available dance styles courses organized per dance school: first all the courses of the school *CityDance*, and next the ones of *DanceCenter*.

Content of *schoolcourses.txt* after execution:

STYLE	LEVEL	TEACHER	SCHOOL	PRICE
hiphop	beginner	Perez	CityDance	70
salsa	advanced	Diaz	CityDance	80
breakdance	beginner	Montoya	CityDance	75
ballet	junior	Mojica	CityDance	75
contemporary	intermediate	Tormo	CityDance	85
salsa	advanced	Diaz	CityDance	75
contemporary	beginner	Aguilar	DanceCenter	70
flamenco	intermediate	Sanchez	DanceCenter	85
hiphop	intermediate	Benavides	DanceCenter	80
tapdance	beginner	Montero	DanceCenter	60
hiphop	beginner	Benavides	DanceCenter	75
salsa	intermediate	Garcia	DanceCenter	85

SOLUTION

```
clear;
clc;
fid = fopen('schools.txt', 'rt');
fid2 = fopen('schoolcourses.txt', 'wt');
if (fid < 0) || (fid2 <0)
    disp('Error opening file');
else
    %READ FILE AND STORE CONTENTS
    centers = {'CityDance', 'DanceCenter'};
    i = 0;
    firstline = fgets(fid);
    while (feof(fid) ==0)
        i = i + 1;
        info = textscan(fid, '%s %s %s %s %d', 1);
        schools(i).style = info{1}{1};
        schools(i).level = info{2}{1};
        schools(i).teacher = info{3}{1};
        schools(i).school = info{4}{1};
        schools(i).price = info{5}(1);
    end
end
fclose(fid);
%DISPLAY AVAILABLE STYLES
disp('The dance styles available are:');
fprintf(fid2,'STYLE\tLEVEL\tTEACHER\tSCHOOL\tPRICE');
for i = 1:2
    fprintf('IN %s:\n', upper(centers{i}));
    for j = 1:length(schools)
        if strcmpi(centers{i}, schools(j).school)
            fprintf('%s - %s - price: %d\n', schools(j).style,
schools(j).level, schools(j).price);
            fprintf(fid2,'\n%s\t%s\t%s\t%s\t%d', schools(j).style,
schools(j).level, schools(j).teacher, schools(j).school,
schools(j).price);
        end
    end
end
fclose(fid2);
disp('File created successfullly!');
```

Exercise 13

Modify the previous program so that the program saves the list with courses organized by school and by level (first junior courses, next beginner, next intermediate and next advanced), as in the following example:

Content of *schoolcourses.txt* after execution:

```
STYLE          LEVEL         TEACHER   SCHOOL        PRICE
ballet         junior        Mojica    CityDance     75
hiphop         beginner      Perez     CityDance     70
breakdance     beginner      Montoya   CityDance     75
contemporary   intermediate  Tormo     CityDance     85
salsa          advanced      Diaz      CityDance     75
contemporary   beginner      Aguilar   DanceCenter   70
tapdance       beginner      Montero   DanceCenter   60
hiphop         beginner      Benavides DanceCenter   75
```

flamenco	intermediate	Sanchez	DanceCenter	85
hiphop	intermediate	Benavides	DanceCenter	80
salsa	intermediate	Garcia	DanceCenter	85

SOLUTION

```
clear;
clc;
fid = fopen('schools.txt', 'rt');
fid2 = fopen('schoolcourses.txt', 'wt');
if (fid < 0) || (fid2 <0)
    disp('Error opening file');
else
    %READ FILE AND STORE CONTENTS
    centers = {'CityDance', 'DanceCenter'};
    levels = {'junior', 'beginner', 'intermediate', 'advanced'};
    i = 0;
    firstline = fgets(fid);
    while (feof(fid) ==0)
        i = i + 1;
        info = textscan(fid, '%s %s %s %s %d', 1);
        schools(i).style = info{1}{1};
        schools(i).level = info{2}{1};
        schools(i).teacher = info{3}{1};
        schools(i).school = info{4}{1};
        schools(i).price = info{5}(1);
    end
end
fclose(fid);
%DISPLAY AVAILABLE STYLES
disp('The dance styles available are:');
fprintf(fid2,'STYLE\tLEVEL\tTEACHER\tSCHOOL\tPRICE');
for i = 1:2
    fprintf('IN %s:\n', upper(centers{i}));
    for k = 1:length(levels)
        fprintf('LEVEL %s:\n', upper(levels{k}));
        for j = 1:length(schools)
            if (strcmpi(centers{i}, schools(j).school) == 1) &&
(strcmpi(levels{k}, schools(j).level) == 1)
                fprintf('%s - %s - price: %d\n', schools(j).style,
schools(j).level, schools(j).price);
                fprintf(fid2,'\n%s\t%s\t%s\t%s\t%d', schools(j).style,
schools(j).level, schools(j).teacher, schools(j).school,
schools(j).price);
            end
        end
    end
end
fclose(fid2);
disp('File created successfullly!');
```