Computer Programming
Bachelor in Biomedical Engineering
Bachelor in Applied Mathematics and Computing

Exercise Sheet 9

Cell Arrays and Structures

- SOLUTIONS -

Content Table

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

Exercise 1

Write a program that asks the user to introduce names and ages of students until the user introduces an empty name. Next, the program asks to introduce numbers and prints the corresponding name and age following the order of introduction. The program ends when the user introduces a 0. **Solve this exercise using structures**.

Example of execution:

```
Introduce a name: Pedro
Introduce age: 20
Introduce a name: Ana
Introduce age: 18
Introduce a name: Elena
Introduce age: 19
Introduce a name:
You’ve finished introducing names.
Introduce a number: 2
Ana: 18 years old
Introduce a number: 3
Elena: 19 years old
Introduce a number: 0
Ok, bye!
```

SOLUTION

```
clear;
varName = input('Introduce a name: ', 's');
cont = 0;
while (isempty(varName) == 0)
    varAge = input('Introduce an age: ');
    cont = cont + 1;
    student(cont).name = varName;
    student(cont).age = varAge;
    varName = input('Introduce a name: ', 's');
end
disp('You''ve finished introducing names');
varNumber = input('Introduce a number: ');
while (varNumber ~= 0)
    if (varNumber > cont)
        disp ('No names matching that number.');
    else
        fprintf('%s: %d years old', student(varNumber).name,
student(varNumber).age);
    end
    varNumber = input('\nIntroduce a number: ');
end
disp('Ok, bye!');
```

Exercise 2

a) Modify the previous program so that after finishing introducing the names, the program asks the user to introduce names and prints the corresponding age. The program ends when the user introduces an empty name. You can assume that there are no students with the same name. **Solve this exercise using structures.**

Example of execution:

```
Introduce a name: Pedro
Introduce age: 20
Introduce a name: Ana
Introduce age: 18
Introduce a name: Elena
Introduce age: 19
Introduce a name:
You’ve finished introducing names.
Introduce a name: Pedro
Pedro is 20 years old
Introduce a name: Ana
Ana is 18 years old
Introduce a name:
Ok, bye!
```

SOLUTION

```
clear;
varName = input('Introduce a name: ', 's');
cont = 0;
while (isempty(varName) == 0)
    varAge = input('Introduce an age: ');
    cont = cont + 1;
    student(cont).name = varName;
    student(cont).age = varAge;
    varName = input('Introduce a name: ', 's');
end
disp('You''ve finished introducing names');
varName = input('Introduce a name: ', 's');
while (isempty(varName) == 0)
    cont = 1;
    bFound = 0;
    age = -1;
    while ((cont <=length(student)) && (bFound == 0))
        if (strcmp(student(cont).name, varName) == 1)
            age = student(cont).age;
            bFound = 1;
        else
            cont = cont + 1;
        end
    end
    if (bFound == 0)
        disp ('Sorry, nobody with that name.');
    else
```

```
        fprintf('%s is %d years old\n', varName, age);
    end
    varName = input('Introduce a name: ', 's');
end
disp('Ok, bye!');
```

b) Modify the previous program so that the search for the name of the student in the vector of structures containing the information is done with a different function. More specifically, the function should receive the vector of structures and the name to look for, and return the corresponding age of the student with that name. If the name cannot be found the function should return the value 0.

MAIN PROGRAM

```
clear;
varName = input('Introduce a name: ', 's');
cont = 0;
while (isempty(varName) == 0)
    varAge = input('Introduce an age: ');
    cont = cont + 1;
    student(cont).name = varName;
    student(cont).age = varAge;
    varName = input('Introduce a name: ', 's');
end
disp('You''ve finished introducing names');
varName = input('Introduce a name: ', 's');
while (isempty(varName) == 0)
    age = findAge(student, varName);
    if (age == 0)
        disp('Sorry, nobody with that name.');
    else
        fprintf('%s is %d years old\n', varName, age);
    end
    varName = input('Introduce a name: ', 's');
end
disp('Ok, bye!');
```

FUNCTION

```
function [rdo] = findAge(ststudent, aName)
cont = 1;
bFound = 0;
rdo = 0;
while ((cont <= length(ststudent)) && (bFound == 0))
    if (strcmp(ststudent(cont).name, aName) == 1)
        rdo = ststudent(cont).age;
        bFound = 1;
    else
        cont = cont +1;
    end
end
end
```

Exercise 3

Modify the previous program so that for each student the program asks to introduce his/her results of the last 3 exams. At the end, the program asks for names of students, and displays their results, the mean and the difference to the mean of all the results of the class. **Solve this exercise using structures**.

Example of execution:

```
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
 Exam results: 9.00 10.00 8.00
 Mean: 9.00
 Difference to class mean: 0.11

 Introduce a name: Ana
 Exam results: 7.00 8.00 9.00
 Mean: 8.00
 Difference to class mean: -0.89

 Introduce a name:
 Ok, bye!
```

MAIN PROGRAM

```
clear;
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');
varName = input('\nIntroduce a name: ', 's');
while (isempty(varName) == 0)
    posStud = findStudent(students, varName);
    if (posStud == 0)
        disp('Sorry, nobody with that name.');
    else
        fprintf('Exam results: ');
        for j=1:3
            fprintf('%.2f ', students(posStud).results(j));
        end
        fprintf('\nMean: %.2f', students(posStud).mean);
        fprintf('\nDifference to class mean: %.2f', students(posStud).mean-
classMean);
    end
    varName = input('\nIntroduce a name: ', 's');
end
disp('Ok, bye!');
```

FUNCTION

```
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
```

Exercise 4

Create a structure ‘time’ for storing hours, minutes and seconds. Write a program that asks the user to introduce times in the standard format of the runners of a race, until he or she introduces an empty one. The program stores them in a vector of structures, and then prints the time corresponding to the fastest and slowest runner.

Example of execution:

```
Introduce a time: 01:59:45
Introduce a time: 02:02:34
Introduce a time: 01:12:56
Introduce a time:
Fastest: 01:12:56
Slowest: 02:02:34
```

Note: At some point during this exercise you will need to transform a string containing numbers (i.e. ‘59’) into a real number (59). You can do that using the function *str2num*. This function receives a string containing numbers and returns the corresponding number: e.g. str2num(‘59’) returns 59.

SOLUTION

```
clear;
cont = 0;
sTime = input('Introduce a time: ','s');
while ~isempty(sTime)
    sHours = sTime(1:2);
    sMinutes = sTime(4:5);
    sSeconds = sTime(7:8);
    cont = cont +1;
    time(cont).hours = str2num(sHours);
    time(cont).minutes = str2num(sMinutes);
    time(cont).seconds = str2num(sSeconds);
    time(cont).sformat = sTime;
    sTime = input('Introduce a time: ','s');
end

indexMinTime = 1;
minSeconds = time(1).seconds + time(1).minutes*60 + time(1).hours * 3600;

indexMaxTime = 1;
maxSeconds  = time(1).seconds + time(1).minutes*60 + time(1).hours * 3600;

for i=1:cont
    totalSeconds = time(i).seconds + time(i).minutes*60 + time(i).hours *
3600;
    if totalSeconds < minSeconds
        indexMinTime = i;
        minSeconds = totalSeconds;
    end
    if totalSeconds > maxSeconds
        indexMaxTime = i;
        maxSeconds = totalSeconds;
    end
end

fprintf('Fastest: %s\n',time(indexMinTime).sformat);
fprintf('Slowest: %s\n',time(indexMaxTime).sformat);
```

Exercise 5

Write a program that asks the user to introduce coordinates x and y of different points. Next, the program prints the information of all the points introduced one after the other.

Example of execution:

```
Introduce the x coordinate: 1.5
Introduce the y coordinate: 2.5
Do you want to introduce more points (Y/N)? Y
Introduce the x coordinate: 3
Introduce the y coordinate: 5.12
Do you want to introduce more points (Y/N)? Y
Introduce the x coordinate: 2
Introduce the y coordinate: 2
Do you want to introduce more points (Y/N)? N
The coordinates of the points are:
Point 1: (1.50, 2.50)
Point 2: (3.00, 5.12)
Point 3: (2.00, 2.00)
```

SOLUTION

```
clear;
more = 'Y';
i = 0;
while (more == 'Y')
    i = i + 1;
    point(i).x = input('Introduce the x coordinate: ');
    point(i).y = input('Introduce the y coordinate: ');
    more = input('Do you want to introduce more points (Y/N)? ','s');
end
disp ('The coordinates of the points are: ');
for j = 1:i
    fprintf('Point %d: (%.2f, %.2f)\n', j, point(j).x, point(j).y);
end
```

Exercise 6

Modify the previous program so that after introducing the points the user is asked to introduce a number. The program will print the distances from the point selected by the user (according to their order of introduction) to the rest of the points on screen. Finally, the program displays which point is furthest from the one selected by the user.

Example of execution:

```
Introduce the x coordinate: 1.5
Introduce the y coordinate: 2.5
Do you want to introduce more points (Y/N)? Y
Introduce the x coordinate: 3
Introduce the y coordinate: 5.12
Do you want to introduce more points (Y/N)? Y
Introduce the x coordinate: 2
Introduce the y coordinate: 2
Do you want to introduce more points (Y/N)? N
Introduce the number of one of the points: 2
The distances to point (3.00, 5.12) are:
Point 1: distance 3.02
Point 3: distance 3.28
The closest point is point (1.50, 2.50)
```

MAIN PROGRAM

```
clear;
% Introduce the points
count = 0;
more = 'Y';
while (more == 'Y')
    count = count + 1;
    point(count).x = input('Introduce the x coordinate: ');
    point(count).y = input('Introduce the y coordinate: ');
    more = input('Do you want to introduce more points (Y/N)? ','s');
end

numP = input('Introduce the number of one of the points: ');
fprintf('The distances to point (%.2f,%.2f) are: ',
point(numP).x,point(numP).y);

% Compute the distances to the other points
closestPoint = -1;
for i = 1:count
    if (i ~= numP)
        distance = pointsDistance(point(i),point(numP));
        fprintf('\nPoint %d: distance %.2f ', i, distance);
        if closestPoint < 0 || distance < minDistance
            minDistance = distance;
            closestPoint = i;
        end
    end
end

fprintf('\nThe closest point is point (%.2f,%.2f)\n',
point(closestPoint).x, point(closestPoint).y);
```

FUNCTION

```
function [distance] = pointsDistance(pointA, pointB)
% Function that receives two points and returns the distance between them
distance = sqrt((pointA.x-pointB.x)^2+ (pointA.y-pointB.y)^2);
end
```

Exercise 7

Write a program for a painting application, based on the structure 'point' we defined in the last exercise. The program allows the user to define rectangles (defined by four points) and circles (defined by a point and a radius), and allows to specify their colors. Once the user finishes introducing the information, the program list the rectangles and the circles introduced.

Note: Keep in mind that a structure can be used as part of (as a field) of another structure.

Example of execution:

```
Do you want to introduce a rectangle (r), circle (c), or finish
(e)? c
Introduce the x coordinate of the center of the circle: 0
Introduce the y coordinate of the center of the circle: 0
Introduce the radius of the circle: 2
Introduce the color of the circle: blue
Do you want to introduce a rectangle (r), circle (c), or finish
(e)? r
Point 1:
Introduce the x coordinate:  1
Introduce the y coordinate:  1
Point 2:
Introduce the x coordinate:  1
Introduce the y coordinate:  -1
Point 3:
Introduce the x coordinate:  -1
Introduce the y coordinate:  -1
Point 4:
Introduce the x coordinate:  -1
Introduce the y coordinate:  1
Introduce the color of the rectangle: black
Do you want to introduce a rectangle (r), circle (c), or finish
(e)? c
Introduce the x coordinate of the center of the circle: 3
Introduce the y coordinate of the center of the circle: 2
Introduce the radius of the circle: 1.25
Introduce the color of the circle: red
Do you want to introduce a rectangle (r), circle (c), or finish
(e)? e

** FIGURES INTRODUCED **
 Rectangle Nr 1: (1.00,1.00)-(1.00,-1.00)-(-1.00,-1.00)-(-
1.00,1.00) BLACK
 Circle Nr 1: (0.00,0.00) 2.00 BLUE
 Circle Nr 2: (3.00,2.00) 1.25 RED
```

SOLUTION

```
clear;
countR  = 0;
countC  = 0;
more = input('Do you want to introduce a rectangle (r), circle (c), or
finish (e)? ','s');
while ((more ~= 'e') && (more ~= 'E'))
    switch more
        case {'r', 'R'}
            countR = countR + 1;
            for i=1:4
                fprintf('Point %d: \n', i);
                rectangle(countR).point(i).x = input('Introduce the x
coordinate: ');
                rectangle(countR).point(i).y = input('Introduce the y
coordinate: ');
            end
            rectangle(countR).color = input('Introduce the color of the
rectangle: ','s');
        case {'c', 'C'}
            countC = countC + 1;
            circle(countC).point.x = input('Introduce the x coordinate of
the center of the circle: ');
            circle(countC).point.y = input('Introduce the y coordinate of
the center of the circle: ');
            circle(countC).radius = input('Introduce the radius of the
circle: ');
            circle(countC).color = input('Introduce the color of the
circle: ','s');
    end
    more = input('Do you want to introduce a rectangle (r), circle (c), or
finish (e)? ','s');
end % This is the end of the while loop
disp('** FIGURES INTRODUCED **');
% First we print the info about the rectangles
for i=1:countR
    fprintf('Rectangle Nr %d: (%.2f,%.2f)-(%.2f,%.2f)-(%.2f,%.2f)-
(%.2f,%.2f) %s\n', i, rectangle(i).point(1).x, rectangle(i).point(1).y,
rectangle(i).point(2).x, rectangle(i).point(2).y, rectangle(i).point(3).x,
rectangle(i).point(3).y, rectangle(i).point(4).x, rectangle(i).point(4).y,
upper(rectangle(i).color));
end
% Then we print the info about the circles
for j=1:countC
fprintf('Circle Nr %d: (%.2f,%.2f) %.2f %s\n', j, circle(j).point.x,
circle(j).point.y, circle(j).radius, upper(circle(j).color));
end
```

Exercise 8

Modify the previous program so the figures are listed in the same order as they were introduced. Note: Use cell arrays to solve this exercise.

Example of execution:

```
Do you want to introduce a rectangle (r), circle (c), or finish
(e)? c
Introduce the x coordinate of the center of the circle: 0
Introduce the y coordinate of the center of the circle: 0
Introduce the radius of the circle: 2
Introduce the color of the circle: blue
Do you want to introduce a rectangle (r), circle (c), or finish
(e)? r
 Point 1:
Introduce the x coordinate:  1
Introduce the y coordinate:  1
 Point 2:
Introduce the x coordinate:  1
Introduce the y coordinate:  -1
Point 3:
Introduce the x coordinate:  -1
Introduce the y coordinate:  -1
 Point 4:
Introduce the x coordinate:  -1

Introduce the y coordinate: 1
Introduce the color of the rectangle: black
Do you want to introduce a rectangle (r), circle (c), or finish
(e)? c
Introduce the x coordinate of the center of the circle: 3
Introduce the y coordinate of the center of the circle: 2
Introduce the radio of the circle: 1.25
Introduce the color of the circle: red
Do you want to introduce a rectangle (r), circle (c), or finish
(e)? e

** FIGURES INTRODUCED **
Figure Nº 1 - Circle : (0.00,0.00) 2.00 BLUE
Figure Nº 2 - Rectangle : (1.00,1.00)-(1.00,-1.00)-(-1.00,-1.00)-
(-1.00,1.00) BLACK
Figure Nº 3 - Circle: (3.00,2.00) 1.25 RED
```

SOLUTION

```
% When using cell arrays, don't forget to clear them
clear cellFigures;
clear rectangle;
clear circle;
count = 0;
more = input('Do you want to introduce a rectangle (r), circle (c), or
finish (e)? ','s');
while ((more ~= 'e') && (more ~= 'E'))
    switch more
        case {'r', 'R'}
            for i=1:4
                fprintf('Point %d: \n', i);
                rectangle.point(i).x = input('Introduce the x coordinate:
');
```

```
                rectangle.point(i).y = input('Introduce the y coordinate: 
');
            end
            rectangle.color = input('Introduce the color of the rectangle: 
','s');
            count = count + 1;
            cellFigures{count,1} = rectangle;
            cellFigures{count,2} = 'R';
        case {'c', 'C'}
            circle.point.x = input('Introduce the x coordinate of the 
center of the circle: ');
            circle.point.y = input('Introduce the y coordinate of the 
center of the circle: ');
            circle.radius = input('Introduce the radius of the circle: ');
            circle.color = input('Introduce the color of the circle: 
','s');
            count = count + 1;
            cellFigures{count,1} = circle;
            cellFigures{count,2} = 'C';
    end
    more = input('Do you want to introduce a rectangle (r), circle (c), or 
finish (e)? ','s');
end % this is the end of the while loop
disp('** FIGURES INTRODUCED **');
for i=1:count
    switch cellFigures{i,2}
        case 'R'
            rectangle = cellFigures{i,1};
            fprintf('Figure Nr %d - Rectangle : (%.2f,%.2f)-(%.2f,%.2f)-
(%.2f,%.2f)- (%.2f,%.2f) %s\n', i, rectangle.point(1).x, 
rectangle.point(1).y, rectangle.point(2).x, rectangle.point(2).y, 
rectangle.point(3).x, rectangle.point(3).y, rectangle.point(4).x, 
rectangle.point(4).y, upper(rectangle.color));
        case 'C'
            circle = cellFigures{i,1};
            fprintf('Figure Nr %d - Circle : (%.2f,%.2f) %.2f %s\n', i, 
circle.point.x, circle.point.y, circle.radius, upper(circle.color));
    end
end
```