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

Course 2020 / 2021

Exercise Sheet – Preparing Midterm Exam 1
–SOLUTIONS-

Content Table

Exercise 1

Write a program that computes the average of a series of numbers that the user introduces:

- First, it asks the user how many numbers he/she wants to introduce.
- Next, it prints the average of the POSITIVE numbers introduced by the users (negative numbers are ignored) on screen

Example of execution:

```
How many numbers do you want to introduce? 3
Introduce a number: 7
Introduce a number: -2
Introduce a number: 8
The average is 7.5
```

```
clear;
maxnum = input ('How many numbers do you want to introduce? ');
nsum = 0;
npos = 0;
for i = 1:maxnum
    num = input ('Introduce a number: ');
    if num >= 0
        npos = npos +1;
        nsum = nsum + num;
    end
end
average = nsum / npos;
fprintf('The average value is %.1f\n', average);
```

Exercise 2

Write a program that computes a mathematical operation on a series of numbers introduced by the user. The program asks the user to introduce how many numbers he/she wants to introduce and the operation he/she wants to perform (sum or multiplication). Then it asks to introduce the numbers and prints the result of the operation.

Example of execution:

```
How many numbers do you want to introduce? 4
Which operation do you want to perform (Sum(s) / Multiplication (m))? s
Introduce an element: 2
Introduce an element: 4
Introduce an element: 6
Introduce an element: 1

The result of the operation is 13
```

Another example of execution:

```
How many numbers do you want to introduce?  5
Which operation do you want to perform (Sum(s) / Multiplication (m))? m
Introduce an element: 2
Introduce an element: 2
Introduce an element: 3
Introduce an element: 3
Introduce an element: 4

The result of the operation is 144
```

```
vectsize = input('How many numbers do you want to introduce? ');
opt = input('Which operation do you want to perform? Sum(s) /
Multiplication (m) ', 's');
vect = zeros(1,vectsize);

switch opt
    case 's'
        result = 0;
    case 'm'
        result = 1;
end

for i = 1:vectsize
    vect(i) = input('Introduce an element: ');
    switch opt
        case 's'
            result = result + vect(i);
        case 'm'
            result = result * vect(i);
    end
end

fprintf('The result of the operation is %d\n', result);
```

Exercise 3

Write a program that asks the user to introduce a series of numbers, and prints the result of multiplying the sum of the even ones by the sum of the odd ones on screen. The output of the program should look like this:

Example of execution:

```
How many numbers do you want to introduce? 5
Introduce an element: 2
Introduce an element: 7
Introduce an element: 3
Introduce an element: 6
Introduce an element: 2

The result is 100
```

```
vectsize = input('How many numbers do you want to introduce? ');
vect = zeros(1,vectsize);
sumod = 0;
sumev = 0;
for i = 1:vectsize
    vect(i) = input('Introduce an element: ');
    if(rem(vect(i),2) == 0)
        sumev = sumev + vect(i);
    else
        sumod = sumod + vect(i);
    end
end
fprintf('The result is %d\n', sumev * sumod);
```

Exercise 4

Write a program that asks the user to introduce four numbers. The program shows the number of values between the two first numbers (these numbers are included) that can be divided by the other two on screen.

Example of execution:

```
Introduce a value: 2
Introduce another value: 25
Introduce a third one: 3
Introduce a fourth on: 4

There are 2 values between 2 and 25 that can be divided by 3 and 4
```

```
valmin  = input('Introduce a value: ');
valmax  = input('Introduce another value: ');
valdiv1 = input('Introduce a third one: ');
valdiv2 = input('Introduce a fourth one: ');

count = 0;
for i = valmin:valmax
    if ((rem(i,valdiv1) == 0) && (rem(i,valdiv2) == 0))
        count = count +1;
    end
end
fprintf('There are %d values between %d and %d that can be divided by %d
and %d\n', count, valmin, valmax, valdiv1, valdiv2);
```

Exercise 5

Write a program that asks the user to introduce a number (n), and then computes the n numbers of the Fibonacci sequence. Each number is computed by summing the two preceding ones, and the sequence always starts with with 0 and 1: 0, 1, 1, 2, 3, 5, 8, 13, ...

Solve the exercise without vectors.

Example of execution:

```
Introduce the numbers of Fibonacci numbers to print: 10
0 1 1 2 3 5 8 13 21 34
```

```
clear;
n = input('Introduce the numbers of Fibonacci numbers to print: ');

if n >= 1
    fprintf(' 0 ');
end
if n >= 2
    fprintf(' 1 ');
end

fibAnt2 = 0;
% we use this variable to store the value of the n-2 number of the
% fibonacci sequence
fibAnt1 = 1;
% we use this variable to store the value of the n-1 number of the
% fibonacci sequence
```

```
for i = 3:n
% we save the value of the n numbers of the fibonacci sequence
    fib = fibAnt2 + fibAnt1;
    fprintf(' %d ', fib);
    fibAnt2 = fibAnt1;
    fibAnt1 = fib;
end
```

Exercise 6

Write a program that asks the user to introduce a number and creates a square matrix with that number of lines and columns. The elements of the matrix will go from 1 until num_ lines * num_ columns.

Example of execution:

```
Introduce a number: 5

1      2      3      4      5

6      7      8      9      10

11     12     13     14     15

16     17     18     19     20

21     22     23     24     25
```

```
clear;
n = input('Introduce a number: ');
A = zeros(n,n);
len = n*n;
row = 1;
col = 0;
for i = 1:len
    col = col + 1;
    if (col > n)
        col = 1;
        row = row + 1;
    end
    A(row, col) = i;
end
disp(A);
```

Exercise 7

Write a program that

- First asks the user how many numbers he/she wants to introduce
- Prints the sum of the numbers introduced in groups of 3
- Prints the value of the biggest sum previously obtained

Note: if the number of values introduced by the user is not a multiple of 3 the program prints the sum of the values that are left over at the end of the series.

Example of execution:

```
How many numbers do you want to introduce? 6
Introduce a number: 1
Introduce a number: 2
Introduce a number: 4
The sum is 7
Introduce a number: 2
Introduce a number: 1
Introduce a number: 5
The sum is 8
The biggest sum is 8
```

```
clear;
maxnum = input ('How many numbers do you want to introduce? ');
nsum    = 0;
nmaxsum = 0;
ncont   = 0;
val1    = 0;
val2    = 0;
val3    = 0;
for i = 1:maxnum
    num = input ('Introduce a number: ');
    ncont = ncont + 1;
    if (ncont == 1)
        val1 = num;
    elseif (ncont == 2)
        val2 = num;
    elseif (ncont == 3)
        val3 = num;
        nsum = val1 + val2 + val3;
        fprintf('The sum is %d\n',nsum);
        if nsum > nmaxsum
            nmaxsum = nsum;
        end
        ncont = 0;
        val1 = 0;
        val2 = 0;
        val3 = 0;
    end
end

if (ncont ~= 0) % this means that there are some values left over
    nsum = val1 + val2 + val3;
    fprintf('The sum is %d\n', nsum);
    if nsum > nmaxsum
        nmaxsum = nsum;
    end
end
fprintf('The biggest sum is %d\n',nmaxsum);
```

Exercise 8

Write a program that allows the user to insert a new number in a vector of numbers, at any desired position, without removing any of the already existing values. To do this, it will ask the user to introduce a value and the position where it will be stored. You can start with the following already initialized vector: A = [1,2,3,4,5,6,7,8,9];

Note: you should solve the exercise using some kind of loop.

Example of execution:

```
Introduce new value: 0
Introduce its position: 5
The new vector is 1 2 3 4 0 5 6 7 8 9
```

```
clear;
A = [1,2,3,4,5,6,7,8,9];
val = input ('Introduce new value: ');
pos = input ('Introduce its position: ');
if pos < length(A)
    vectaux = [];
    for i = 1:length(A)+1
        if i < pos
            vectaux = [vectaux A(i)];
        elseif i == pos
            vectaux = [vectaux val];
        else
            vectaux = [vectaux A(i-1)];
        end
    end
    A = vectaux;
else
    A = [A val];
end
disp('The new vector is: ');
fprintf('%d ', A);
```

Exercise 9

Write a program for a new type of ATM that gives the user the money in the notes he/she prefers. The program should work like this: first, it asks the user a quantity in euros, and next it asks the user to specify a type of note. The ATM will dispense the note chosen, and next it asks the user to introduce another type of note, repeating the process until all of the quantity has been dispensed. The program should consider the case where the user asks for a note greater than the quantity that is left to deliver, and should display an error message in this case. The ATM can give notes of 100, 50, 20, 10 and 5 euros.

Example of execution:

```
Introduce an amount of money: 100
What kind of bill do you wish to withdraw? 50
You withdrew 50 euros.
You have 50 euros left.
What kind of bill do you wish to withdraw? 100
Error. Not enough money left to dispense this note
You have 50 euros left.
What kind of bill do you wish to withdraw? 20
You withdrew 20 euros.
You have 30 euros left.
What kind of bill do you wish to withdraw? 10
You withdrew 10 euros.
You have 20 euros left.
What kind of bill do you wish to withdraw? 50
Error. Not enough money left to dispense this note
You have 20 euros left.
What kind of bill do you wish to withdraw? 20
You withdrew 20 euros.
```

You have 0 euros left. Bye!

```
clear;
money = input('Introduce an amount of money: ');
while money > 0
    withdraw = input('What kind of bill do you wish to withdraw? ');
    if (money-withdraw) < 0
        disp ('Error. Not enough money left to dispense this note');
        fprintf('You have %d euros left.\n',money);
    else
        fprintf('You withdrew %d euros.',withdraw);
        money = money - withdraw;
        fprintf('\nYou have %d euros left.\n',money);
    end
end
```

Exercise 10

Given a vector A = [2, 8, 3, 30, 4, 50, 100, 200, 4, 80, 500], write a program that splits this vector into three vectors with different value ranges:

- [0−10]
- [10−100]
- [100−1000]

Prints these three vectors on screen.

Example of execution:

```
Elements in [0,10]:
2 8 3 4 4
Elements in [10,100]:
30 50 80
Elements in [100,1000]:
100 200 500
```

```
clear;
A = [2, 8, 3, 30, 4, 50, 100, 200, 4, 80, 500];
vector1 = [];
vector2 = [];
vector3 = [];

for val=A
    if (val>=0 && val<10)
        vector1 = [vector1 val];
    elseif (val>=10 && val<100)
        vector2 = [vector2 val];
    elseif (val>=100 && val<1000)
        vector3 = [vector3 val];
    end
end
disp('Elements in [0,10]: ');
for value=vector1
    fprintf('%d ',value);
end
fprintf('\nElements in [10,100]: \n');
for value=vector2
    fprintf('%d ',value);
```

```
end
fprintf('\nElements in [100,1000]: \n');
for value=vector3
    fprintf('%d ',value);
end
```

Exercise 11

Write a program that asks the user to introduce numbers and an operation (+, -,=) between two different numbers, until the user introduces =. In the end, the program must display the result of the operations.

Example of execution:

```
Introduce a number: 1
Choose an operation (+/-/=): +
Introduce a number: 2
Choose an operation (+/-/=): -
Introduce a number: 4
Choose an operation (+/-/=): +
Introduce a number: 7
Choose an operation (+/-/=): =
The result is 6
```

```
res = input('Introduce a number: ');
op = input('Choose an operation (+/-/=) ', 's');
while op ~= '='
    num = input('Introduce a number: ');
    switch op
        case '+'
            res = res + num;
        case '-'
            res = res - num;
    end
    op = input('Choose an operation (+/-/=)' , 's');
end
fprintf('The result is %d\n', res);
```

Exercise 12

Write a program that reads and stores the festive days in a month. The program will ask the user to introduce the month, and then the number of festivities in that month (which cannot exceed the maximum number of days that the month can have). Then, it will ask for the date of those festivities and store them in a vector after checking that the date is valid.

Example of execution:

```
Introduce the month: 12
Introduce the number of festivities: 33
The number of festivities cannot exceed the days of the month
Introduce the number of festivities: 5
Introduce day: 6
Introduce day: 8
Introduce day: 24
Introduce day: 25
Introduce day: 33
The day cannot exceed the days of the month
Introduce day: 31
The festive days are:
 6     8      24     25     31
```

```
clear;
month = input('Introduce the month: ');
while month < 0 || month > 12
    disp('Invalid month.');
    month = input('Introduce the month: ');
end
switch month
    case 2
        maxdays = 28;
    case {4,6,9,11}
        maxdays=30;
    otherwise
        maxdays=31;
end
n = input('Introduce the number of festivities: ');
festivities = zeros(1,n);
while n > maxdays
    disp('The number of festivities cannot exceed the days of the month.');
    n = input('Introduce the number of festivities: ');
end
count = 0;
while count < n
    day = input('Introduce day: ');
    if day > maxdays
        disp('The day cannot exceed the days of the month.');
    else
        count = count + 1;
        festivities(count)=day;
    end
end
disp('The festive days are: ');
for i = 1:n
    fprintf(' %d ',festivities(i));
end
```