Computer Programming

Bachelor in Biomedical Engineering

Bachelor in Applied Mathematics and Computing

Course 2020 / 2021

Exercise Sheet 7

Functions

- Extra exercises -

- SOLUTIONS -

Content Table

Exercise 7

Write a function that receives a number and returns a vector containing the digits of the number.

Then, write a program to test the function. The program asks the user to introduce a number and it prints the digits

```
Introduce a number: 9456
The digits are:
Digit 1 - 6
Digit 2 - 5
Digit 3 - 4
Digit 4 – 9
```

FUNCTION

```
function [vdigits] = digits(numb)
vdigits = [];
while(numb > 0)
    digit = rem(numb,10);
    numb = floor(numb/10);
    vdigits = [vdigits digit];
end
end
```

MAIN PROGRAM

```
clear;
value = input('Introduce a number: ');
disp('The digits are:');
vector = digits(value); % call function digits
for i = 1:length(vector)
    fprintf('Digit %d - %d\n', i, vector(i));
end
```

Exercise 8

Strong numbers are numbers whose sum of the factorial of its digits is equal to the number itself (e.g. 145 = 1! + 4! + 5!).

Write a function that receives a number and returns 1 if the number is strong and 0 otherwise.

Then, write a program that asks the user to introduce numbers one at a time, and displays whether the numbers are strong or not. After checking whether the number is strong, the program asks the user if he/she wants to continue.

Example of execution:

```
Introduce a number: 145
The number is strong
Do you want to continue (Y/N)? x
Sorry, I don't understand
Do you want to continue (Y/N)? Y
Introduce a number: 125
The number is not strong
Do you want to continue (Y/N)? N
```

FUNCTIONS

```
function [fact] = myFactorial (value)
fact = 1;
for i = 1:value
    fact = fact * i;
end
end
```

```
function [isStrong] = strong (num)
vect = digits(num); % call function digits (see Exercise 7)
sumfact = 0;
for val = vect
    valfact = myFactorial(val); % call function myFactorial
    sumfact = sumfact + valfact;
end
if (sumfact == num)
    isStrong = 1;
else

    isStrong = 0;
end
end
```

MAIN PROGRAM

```
clear;
bContinue = 'Y';
while (bContinue == 'Y')
    number = input('Introduce a number: ');
    bStrong = strong(number);
    if bStrong == 1
        disp('The number is strong');
    else
        disp('The number is not strong');
    end
    bContinue = input('Do you want to continue (Y/N)? ','s');
    while ((bContinue ~= 'Y') && (bContinue ~= 'N'))
        disp('Sorry, I don''t understand');
        bContinue = input('Do you want to continue (Y/N)? ','s');
    end
end
```

Exercise 9

Sometimes we might need to write functions that don't return any value. In that case we write them like this:

function [ ] = functionName (…)
…
end

Write a function that receives a number and prints a Pascal triangle (https://en.wikipedia.org/wiki/Pascal's_triangle) with as many lines as the number received.

SOLUTION

```
function [ ] = pascalTriangle(lines)
A = zeros(lines,lines);
A(:,1) = 1;
for i=2:lines
    for n=2:lines
        A(i,n) = A(i-1,n) + A(i-1,n-1);
    end
end
for j=1:lines
    for m=lines-1:-1:j
        fprintf(' ');
    end
    for k=1:lines
        if (A(j,k) ~= 0)
            fprintf('%d ',A(j,k));
        end
    end
    fprintf('\n');
end
end
```