

Tema 2

Nicoleta Radu

Exercitiul 5, Pag. 6

Baza 10	15	0.12	0.375	6	11.74	13.9375
Baza 2	1111		0.011	110	1011.11	1101.1111
Baza 16	F	0.(1EB85)	0.6	6	B.C	D.F

Exercitiul 7, Pag. 7

```
function []=conversie16(x,n)
% conversie x din baza 10 in baza 2
if nargin~=2
error('nr de argumente gresit! Sintaxa este conversie(nr_baza_10,nr_cifre)');
elseif n<0
error('n reprezinta nr de cifre ale lui x si nu poate fi nr negativ');
end
nrNeg=0;
if x<0
nrNeg=1; x=abs(x); % nrNeg = true daca avem un numar x negativ
end
vect=[]; % folosit pentru a scrie numarul din fata virgulei in baza 2
ParteInt=0; ParteFract=0; % ParteInt,ParteFract - variab folosite pentru test
% conversia pentru partea intreaga a lui x
if fix(x)>0 % verificam daca avem parte intreaga la numarul real x
ParteInt=1; % ParteInt = true daca avem un numar x cu parte intreaga
x1=fix(x); %retin in x1 partea intreaga a lui x
while fix(x1) > 0
rest=mod(x1,16); % retin restul impartirii la 2
x2=fix(x1/16); % retin catul lui x1/2
x1=x2; % retin noua valoare pt x1
switch rest
case 0:9
vect=[vect,rest];
otherwise
vect=[vect,dec2hex(rest)];
end
end
cifre_parte_intreaga=flipplr(vect); % simetrie de la st la dr
else
cifre_parte_intreaga=0;
end
i=0; % indicele
```

```

c=[]; % vectorul cu cifre
% x - numar subunitar
% Transformarea unui numar subunitar x, scris in baza 10, in numarul
% x_tilda = 0,c(1),c(2),...,c(n-1), scris in baza 2 se face conform
% algoritmului de mai jos
if (x-fix(x) > 0) % verificam daca avem parte fractionara la numarul real
ParteFract=1; % ParteFract = true daca avem un numar x cu parte fractionara
while (i~=n)
i=i+1;
parte_zecimala=x-fix(x);
x_tilda=parte_zecimala;
x_tilda=x_tilda*16;
c=[c,fix(x_tilda)]; % vectorul cu cifre de 0 si 1 (cu partile intregi)
x=x_tilda-fix(x_tilda);
end
cifre_parte_zecimala=c;
end
% afisare rezultat
% rezultatul este afisat exact cu n cifre
if n < length(cifre_parte_intreaga)
fprintf('Atentie! n < %d (nr necesar de
cifre)\n',length(cifre_parte_intreaga));
error('nu se poate scrie numarul normalizat! Dati nr de cifre mai mare!')
end
if(ParteInt==1) % exista parte intreaga
fprintf('Conversia in baza 16 pentru n=%d este:\n',n)
if nrNeg==1
fprintf('-'); % pentru ca avem un numar negativ va adauga '-'
end
for i=1:length(cifre_parte_intreaga)
switch class(cifre_parte_intreaga(i))
case 'char'
fprintf('%c',cifre_parte_intreaga(i));
case 'string'
fprintf('%s',cifre_parte_intreaga(i));
otherwise
fprintf('%d',cifre_parte_intreaga(i));
end
end
else
if nrNeg==1
fprintf('-0'); % pentru ca avem un numar negativ va adauga '-'
else
fprintf('0');% daca nu avem valori inainte de virgula
end
end
if(ParteFract==1) % exista parte fractionara
dimensiune_max = length(cifre_parte_zecimala);
if dimensiune_max > 0

```

```

fprintf('.'); % va adauga '.' pentru punctul zecimal
for i=1:dimensiune_max
    switch class(cifre_parte_intreaga(i))
        case 'char'
            fprintf('%c',cifre_parte_intreaga(i));
        case 'string'
            fprintf('%s',cifre_parte_intreaga(i));
        otherwise
            fprintf('%d',cifre_parte_intreaga(i));
    end
end
else
%completam daca este nevoie cu 0 pentru nr de cifre dat
z=zeros(1,n-length(cifre_parte_intreaga));
fprintf('%d',z);
end
else % nu avem parte zecimala
if n ~= length(cifre_parte_intreaga)
% verificam daca este nevoie de punctul zecimal, in cazul nr intregi
fprintf('.');
end
%completam daca este nevoie cu 0 pentru nr de cifre dat
z=zeros(1,n-length(cifre_parte_intreaga));
fprintf('%d',z);
end
fprintf('\n');

```

```

Conversia in baza 16 pentru n=25 este:
2.000000000000000000000000000000

```

```

Conversia in baza 16 pentru n=25 este:
-7F3.7F37F37F37F37F37F37F37F37F37F3

```

```

Conversia in baza 16 pentru n=8 este:
A.AAAAAA

```