

Računske vežbe iz OR2 - Deveti čas

1. Napisati funkcijski m-fajl **karakteri** koji određuje i vraća koliko se u stringu **S** nalazi slova (malih i velikih), koliko cifara i koliko ostalih karaktera. String **S** je ulazni argument funkcijskog fajla, dok su broj slova, cifara i ostalih karaktera izlazni argumenti.

```
function [bs, bc, bo]=karakteri(S);
bs=0; bc=0; bo=0;
for i=1:length(S)
    if (isletter(S(i))==1) % ili (S(i)>='a' & S(i)<='z') | (S(i)>='A' & S(i)<='Z')
        bs=bs+1;
    elseif S(i)>='0' & S(i)<='9'
        bc=bc+1;
    else
        bo=bo+1;
    end
end
```

2. Napisati funkcijski m-fajl pod nazivom **palindrom** čiji je ulazni argument string **S**, i koji ispituje da li je **S** palindrom, odnosno da li se jednako čita sa obe strane. Ukoliko je **S** palindrom fajl vraća broj **1**, a u suprotnom **0**. Primjer palindroma je rečenica 'Ana voli Milovana'. Obratiti pažnju da položaj razmaka ne utiče na donošenje odluke da li je **S** palindrom i da se ne pravi razlika između malih i velikih slova.

```
function a = palindrom(S);
T = [];
for i=1:length(S)
    if (isspace(S(i))==0 %ili if(S(i)~=' ')
        T=[T, S(i)];
    end
end
T=lower(T);
T1=T(length(T):-1:1);
if (strcmp(T, T1)==1)
    a=1;
else
    a=0;
end
```

3. Napisati funkcijski m-fajl **binar** koji za ulazni argument ima prirodan broj **N**, a za izlazni argument ima string **S** koji predstavlja binarni zapis broja **N**.

```
function S = binar(N);
if N<0 | fix(N)~=N
    error('Nije prirodan broj');
end
if N==0
    S='0';
else
    S=[];
    while N~=0
        S=[num2str(rem(N,2)), S];
        N=fix(N/2);
    end
end
```

4. Napisati funkcijski m-fajl **aprox** kojim se aproksimira vrijednost funkcije $y = \sin(x)e^{-x^2}$ na intervalu $[x_1, x_2]$ polinomom $P(x)$, koji predstavlja izlazni argument fajla. Vrijednosti x_1 i x_2 se zadaju kao ulazni argumenti fajla. Red polinoma **n** kojim se vrši aproksimiranje se zadaje kao drugi ulazni argument. Ukoliko se on ne zada podrazumijevati da je $n=4$. Ukoliko se funkcijski fajl poziva sa dva izlazna argumenta, onda kao drugi izlazni argument vratiti maksimalnu apsolutnu vrijednost greške aproksimacije $\varepsilon(x) = \sin(x)e^{-x^2} - P(x)$, na intervalu $[x_1, x_2]$ u 100 tačaka.

```
function [P, max_eps] = aprox(x, n);
if nargin == 1
    n = 4;
end
xp = linspace(x(1), x(2), 10);
yp = sin(xp) .* exp(-xp.^2);
P = polyfit(xp, yp, n);
if nargin == 2
    xs = linspace(x(1), x(2), 100);
    ys1 = sin(xs) .* exp(-xs.^2);
    ys2 = polyval(P, xs);
    max_eps = max(abs(ys1-ys2));
end
```