

Rješenja III i IV zadatka sa završnog ispita

Zadatak 3. Napisati funkcijski fajl **filter3.m** koji za argumente ima niz brojeva **X** i broj **n**. Rezultat funkcije je niz **Y** čija se vrijednost na poziciji **i** računa kao srednja vrijednost **n** elemenata niza **X** iz intervala $\left[X[i - \frac{n-1}{2}], \dots, X[i + \frac{n-1}{2}] \right]$. Za $n=3$, $Y[i]$ se računa kao $Y[i] = \frac{X[i-1] + X[i] + X[i+1]}{3}$. Elemente na pozicijama **i** (na početku i kraju niza) za koje definisani interval nije potpun (nema **n** elemenata), računati kao $Y[i] = X[i]$. Ukoliko je **n** parno prekinuti izvršavanje i javiti poruku greške.

Primjer: Za niz $X=[4 \ 5 \ 3 \ 6 \ 7 \ 2]$ i $n=3$, niz $Y=[4 \ 4 \ 4.67 \ 5.33 \ 5 \ 2]$.

Par mogućih rješenja zadatka:

I predlog rješenja

```
function Y = filter3(X,n)
if(rem(n,2) == 0)
    error('Drugi argument mora biti paran')
end
L = length(X);
Y = zeros(1,L); % znamo da koliko ce elemenata imati Y, pa ga
% inicijalizujemo na nule, da se ne bi mijenjale dimenzije niza
% Y u svakoj iteraciji

Y(1 : (n-1)/2) = X(1 : (n-1)/2); %elementi sa pocetka niza
Y(L - (n-1)/2 + 1 : end) = X(L - (n-1)/2 + 1 : end); %elementi
sa kraja niza
for i = (n-1)/2 + 1 : L - (n-1)/2
    Y(i) = mean(X(i - (n-1)/2 : i + (n-1)/2));
    %2. grupa samo median umjesto mean
end
```

II predlog rješenja

```
function Y = filter3(X,n)
if(rem(n,2) == 0)
    error('Drugi argument mora biti paran')
end
L = length(X);
Y = zeros(1,L); % znamo da koliko ce elemenata imati Y, pa ga
% inicijalizujemo na nule, da se ne bi mijenjale dimenzije niza
% Y u svakoj iteraciji
```

```

for i = 1 : L
    if(i - (n-1)/2 < 1) || (i + (n-1)/2 > L)
        Y(i) = X(i); %elementi sa pocetka i kraja niza
    else
        Y(i) = mean(X(i - (n-1)/2 : i + (n-1)/2));
        % 2. grupa median
    end
end
end

```

Zadatak 4. Napisati program kojim se zadati string **S** mijenja tako što se sva mala slova u najdužoj riječi stringa **S** pretvore u velika. Riječ je niz karaktera između dva razmaka. Smatrati da dva razmaka ne mogu biti jedan do drugog i voditi računa o riječima na početku i kraju stringa.

*Primjer: String **S** = 'Danas radimo završni ispit' postaje **S** = 'Danas radimo ZAVRSNI ispit'.*

Par mogućih rješenja zadatka

I rješenje

```

clear
s = input('Unesite neki string','s');
N = length(s);
s1 = [' ' s ' ']; %postavimo razmak na pocetak i kraj stringa da
% riješimo te granicne slucajeve
poz = find(s1 == ' '); % znamo pozicije razmaka
poz = poz + 1; % sada znamo i pocetak svih rijeci, samo moramo
% imati na umu da posljednji element ukazuje na poziciju N + 2;
d = diff(poz) - 1; % sada imamo vektor sa duzinama rijeci
% 1 smo oduzeli zbog razmaka izmedju rijeci
[m,ind] = max(d); % m je duzina najduze rijeci, ind je redni
% broj te rijeci u unijetom stringu, pa ce poz(ind) biti
% pocetak najduze rijeci
s1(poz(ind):poz(ind)+m-1) = upper(s1(poz(ind):poz(ind)+m-1));
s = s1(2 : end -1); %eliminiseмо dodate razmake
disp(s)

```

rješenje baz komentara

```

clear
s = input('Unesite neki string','s');
N = length(s);
s1 = [' ' s ' '];
poz = find(s1 == ' ');
poz = poz + 1;

```

```

d = diff(poz) - 1;
[m,ind] = max(d);
s1(poz(ind):poz(ind)+m-1) = upper(s1(poz(ind):poz(ind)+m-1));
s = s1(2 : end -1);
disp(s)

```

II rjesenje

```

clear
s = input('Unesite neki string','s');
N = length(s);
poz = find(s == ' '); % znamo pozicije razmaka
poz = poz + 1; % sada znamo i pocetak svih rijeci, osim prve,
% za koju znamo da pocinje od prvog karaktera
poz = [1, poz]; % sada imamo i pocetak prve rijeci
d = diff(poz) -1; % sada imamo vektor sa duzinama pojedinih
% rijeci, osim posljednje
% 1 smo oduzeli zbog razmaka izmedju rijeci
d = [d, N - poz(end) + 1] % N - poz(end) +1 je duzina
% posljednje rijeci
[m,ind] = max(d); % m je duzina najduze rijeci, ind je redni
% broj te rijeci u unijetom stringu, pa ce poz(ind) biti
% pocetak najduze rijeci
s(poz(ind) : poz(ind)+m-1) = upper(s(poz(ind) : poz(ind)+m-1));
disp(s)

```

III rjesenje

```

clear
s = input('Unesite neki string','s');
N = length(s);
duzMax = 0; %duzina maksimalne rijeci
pozMax = 0; % pozicija najduze rijeci
i = 1;
while(i <= N)
    j = i; % i pamti pocetak trenutne rijeci
    while (s(j) ~= ' ' && j < N)
        j = j + 1;
    end
    if j == N
        duz = N - i + 1;
    else
        duz = j - i; %j je pozicija razmaka nakon rijeci
    end
    if duz > duzMax

```

```
        duzMax = duz;  
        pozMax = i;  
    end  
    i = j + 1; %pocetak naredne rijeci  
    % ili izlaz iz petlje ako smo dosli do kraja  
end  
s(pozMax:pozMax+duzMax-1) = upper(s(pozMax:pozMax+duzMax-1));  
disp(s)
```