

Gyakorlófeladatok III.

1. Készítsünk olyan függvényt, amely megadja egy mátrix első és utolsó sorát (két kimenő vektora lesz).
2. Írjunk scriptet, amely bekér a felhasználótól két számot, és közülük kiválasztja a kisebbet.
3. Írjunk scriptet, amely megadja egy beolvasott szám előjelét. (Ezt teszi a beépített `sign` függvény is.)
4. Írjunk programot, amely bekéri egy pont koordinátáit és egy kör adatait a síkon, és eldönti, hogy hol helyezkedik el a pont a körvonalhoz képest.
5. Írjunk függvényt, amely kiszámítja az első n egész szám négyzetösszegét.

Megoldások

1. `elso_utolso.m`

```
function [x y] = elso_utolso(A)
x = A(1,:)
y = A(end,:)
```

2. `minimum.m`

```
% Két szám minimumának a kiszámítása
a = input('Az elso szam:')
b = input('A masodik szam:')
if a < b
m = a
else m = b
end
```

3. `elojel.m`

```
% Egy szam elojelenek megadasa
x = input('Kerem a szamot!')
if x > 0
```

```

y = 1
elseif x == 0
y = 0
else
y = -1
end

```

4. ponteskor.m

```

% Hol helyezkedik el egy pont egy korhoz kepest?
x = input('A pont x koordinataja:')
y = input('A pont y koordinataja:')
u = input('A kor kozeppontjanak x koordinataja:')
v = input('A kor kozeppontjanak y koordinataja:')
r = input('A kor sugara:')
t = sqrt((x-u)^2+(y-v)^2); %A pont es a kor kozepponjanak tavolsaga
if t > r
disp('A pont a koron kivul van.')
elseif t == r
disp('A pont a korvonalon van.')
else
disp('A pont a koron belul van.')
end

```

5. Első megoldás: for ciklussal:

negyzetosszeg1.m

```

function s = negyzetosszeg1(n)
% Az elso n szam negyzetosszegenek a kiszamitasa
s = 0;
for i = 1:n
s = s + i^2;
end

```

Másik megoldás: while ciklussal:

negyzetosszeg2.m

```
function s = negyzetosszeg2(n)
% Az elso n szam negyzetosszegenek a kiszamitasa
s = 0;
i = 1;
while i <= n
s = s+i^2;
i = i+1;
end
```