

```

1: procedure NEKI SORT(A)
2:   n ← length(A)
3:   for i ← 1 to n - 1 do
4:     imin ← i
5:     for j ← i + 1 to n do
6:       if A[j] < A[imin] then
7:         imin ← j
8:       end if
9:     end for
10:    if i ≠ imin then
11:      swap(A[i], A[imin])
12:    end if
13:    writeln(A)
14:  end for
15: end procedure
    
```

- Primeniti algoritam NEKI SORT na ulaz  $A = [5, 1, 2, 3, 4]$  i ispisati stanje niza  $A$  koje se ispisuje u liniji 13.
  - 1, 5, 2, 3, 4
  - 1, 2, 5, 3, 4
  - 1, 2, 3, 5, 4
  - 1, 2, 3, 4, 5
  - 
  -
- Za ulazni niz  $[5, 1, 2, 3, 4]$ , koliko će puta poređenje u liniji 6 biti izvršeno, a koliko puta zamena (swap) u liniji 11?
 

Poređenje: 10 zamena: 4.
- Za obrnuto sortirani ulazni niz  $A$  dužine  $n$ , koliko će puta poređenje u liniji 6 biti izvršeno, a koliko puta zamena (swap) u liniji 11?
 

Poređenje:  $n \cdot (n - 1) / 2$  zamena:  $\lfloor n / 2 \rfloor$ .

4.

Napisati u programskom jeziku C proceduru **transposeA** koja kvadratnu matricu  $A$  formata  $n \times n$  zapisanu u niz po vrstama transponuje u "mestu".

```

void transposeA(double *A,
                int n)
{ // Ovaj kod napisati
  int i, j;
  double temp;
  for (i=0; i<n; i++){
    for (j=i+1; j<n; j++){
      temp = A[i+n+j];
      A[i*n+j]=A[j*n+i];
      A[j*n+i] = temp;
    }
  }
}
    
```

Da li u datom programu za transponovanje može umesto transposeA da se koristi procedura transpose koja transponuje matricu  $A$  formata  $m \times n$  u  $B$  formata  $n \times m$  komandom **transpose(A,A,n,n)**? (Vidi kod desno.) Objasniti:

```

Ne:
Zbog dvostrukog prolaska kroz iste elemente dobio bi se izlaz:
+-----+
| 1.00   2.00  -3.00 |
| 2.00  -7.00 -16.00 |
| -3.00 -16.00 -16.00 |
+-----+
    
```

```

#include <stdio.h>
#include <stdlib.h>

int printmatrix(double*, int, int);
// stampa matricu A formata mXn

void transposeA(double *A, int n)
{
  // Ovaj kod napisati u polje levo
}

void transpose(double *A, double *B, int m, int n)
{ // transpose ( A mXn ) = B nXm
  int i, j;
  for (i=0; i<m; i++){
    for (j=0; j<n; j++){
      B[j*m+i] = A[i*n+j];
    }
  }
}

int main()
{
  double A[]={1,2,-3,4,-7,-16,-7,-18,-16};
  int n = 3;
  printmatrix(A,n,n);
  // transpose(A,A,n,n);
  transposeA(A,n);
  printmatrix(A,n,n);
}
    
```