

# Strukture podataka i algoritmi

## Laboratorijske vježbe 4

### Gradivo s predavanja:

- Vezana lista
- Apstraktni tipovi podataka
- ATP: Red (queue)
- ATP: Stog (stack)
- ATP: Lista (list)

### Objašnjenje zadataka:

- Svaka grupa laboratorijskih vježbi rješava svoje grupu zadataka.
- Uz svaki zadatak zapisana je njegova vrijednost u postocima.
- Potrebno je riješiti zadatke svoje grupe u vrijednosti 50% za prolaz laboratorijskih vježbi.
- U glavnom dijelu programa potrebno je testirati funkcije iz odabranih zadataka

### Zadaci:

#### Grupa L1

1. **(30%)** Napišite funkciju `stack_element_t stack_dequeue (stack_t stack);` koja će dohvatiti i ukloniti najstariji element iz proslijeđenog joj stoga.

Primjer izgleda stoga	Najstariji element	Izgled stoga nakon uklanjanja
3, 10, 6, 7, 1, 9	3	10, 6, 7, 1, 9

2. **(40%)** Napišite funkciju koja primi dvije ATP liste te vrati indeks prvog pojavljivanja druge liste unutar prve, a -1 ukoliko prva lista ne sadrži drugu listu.

Prva lista	Druga lista	Indeks pojavljivanja
0, 3, -1, 1, 3, -1, 5, 9, 7, 8, 9	3, -1, 5	4
0, 3, -1, 1, 3, -1, 5, 9, 7, 8, 9	3, 5, 9	-1

3. **(30%)** Napišite funkciju koja primi jedan ATP red (`typedef int queue_element_t`), iz njega izbací član čija je vrijednost najmanja te kao povratnu vrijednost vrati vrijednost tog izbačenog člana. Ostatak reda ostaje nepromijenjen.

Red	Najmanji	Red nakon izbacivanja
11, 5, 12, 55, 7, 2, 6	2	11, 5, 12, 55, 7, 6

## Grupa L2

1. **(30%)** Napišite funkciju za multipop operaciju nad stogom koja sa zadanog stoga skida  $n$  elemenata (ili dok stog ne ostane prazan) te vraća zadnji skinuti: `stack_element_t stack_multipop(stack_t s, int n);`

Stog	n	Zadnji skinuti
4, 2, 6, 8, 3, 1, 2	3	3
4, 2, 6, 8, 6, 1, 2	9	4

2. **(40%)** Napišite funkciju prototipa `void rotate(list_t list, unsigned distance);` koja izrotira listu u desno za  $distance$  mjesta.

Lista	distance	Lista nakon rotiranja
v, e, l, e, b, i, t	3	b, i, t, v, e, l, e

3. **(30%)** Napišite funkciju koja će primiti dva uzlazno sortirana ATP stoga, te kao povratnu vrijednost vratiti silazno sortiranu listu sačinjenu od unije članova prva dva stoga.

Prvi stog	Drugi stog	Lista
1, 2, 5, 7	2, 3, 6, 8	8, 7, 6, 5, 3, 2, 2, 1

## Grupa L3

1. **(30%)** Napisati funkciju koja će primiti jedan ATP stog, te vratiti 1 ukoliko se u stogu nalazi više parnih brojeva nego neparnih, a 0 inače. Stog treba ostati nepromijenjen.

Stog	Rezultat
4, 2, 6, 8, 6, 1, 2	1
4, 2, 7, 3, 5, 1, 2	0

2. **(30%)** Napišite funkciju koja primi dvije ATP liste, te vrati 1 ukoliko je druga inverzna verzija prve (sadrži obrnuto poredane vrijednosti u članovima), a 0 inače.

Prva lista	Druga lista	Rezultat
4, 2, 6, 8, 6, 1, 2	3, 4, 5, 6, 8	0
4, 2, 6, 8	8, 6, 2, 4	1

3. **(40%)** Napišite funkciju koja primi jednostruko povezanu listu, te je invertira (prvi član postaje posljednji, drugi postaje pretposljednji itd.)

Jednostruko povezana lista	Invertirana lista
12, 13, 2, 5, 7	7, 5, 2, 13, 12

## Grupa L4

1. **(40%)** Definirajte jednostruko povezanu listu, te napišite funkciju koja primi pokazivač na prvi član takve liste, te vrati 1 ukoliko je lista simetrična (prvi član jednak posljednjem, drugi jednak pretposljednem itd.), a 0 inače.

Jednostruko povezana lista	Rezultat
v, e, l, e, b, i, t	0
k, i, s, i, k	1

2. **(30%)** Napisati funkciju koja će primiti jedan ATP stog te vratiti ATP listu u koju će staviti samo parne brojeve sa stoga. Poredak parnih u listi treba biti kakav je bio i u stogu. Poredak u stogu treba ostati nepromijenjen.

Stog	Lista
1, 4, 6, 3, 2, 8	4, 6, 2, 8

3. **(30%)** Napišite funkciju `multi_dequeue` nad redom koja sa zadanog reda skida `n` elemenata (ili dok red ne ostane prazan), te vraća zadnji skinuti:  
`queue_element_t multi_dequeue(queue_t q, int n);`

Red	n	Zadnji skinuti
4, 2, 6, 8, 3, 1, 2	3	6
4, 2, 6, 8, 6, 1, 2	9	2