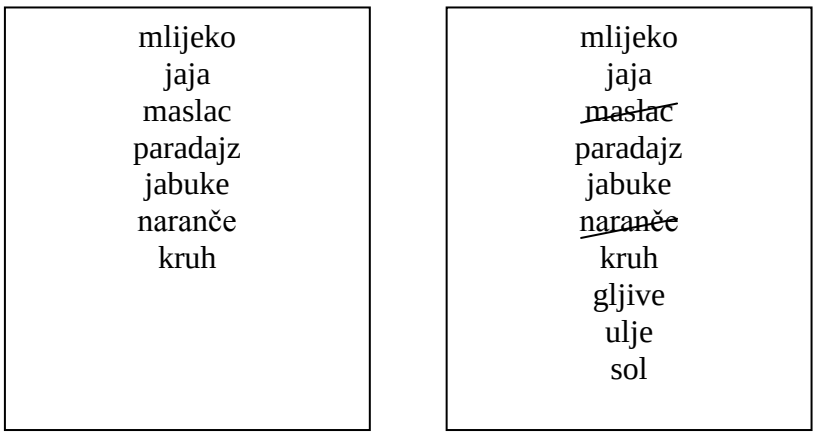
LABORATORIJSKE VJEŽBE IZ ALGORITAMA I STRUKTURA PODATAKA

Vježba 2

Linearne liste

U svakodnevnom govoru izraz "lista" odnosi se na linearni skup podataka. Na slici 3.1. a) prikazana je kupovna lista. Ona sadrži prvi element, drugi element, ..., i zadnji element. Na slici 3.1. b) prikazana je kupovna lista nakon što su dodana tri artikla na kraj liste i izbrisana dva artikla (prikazani precrtano).



Slika 3.1. Kupovna lista

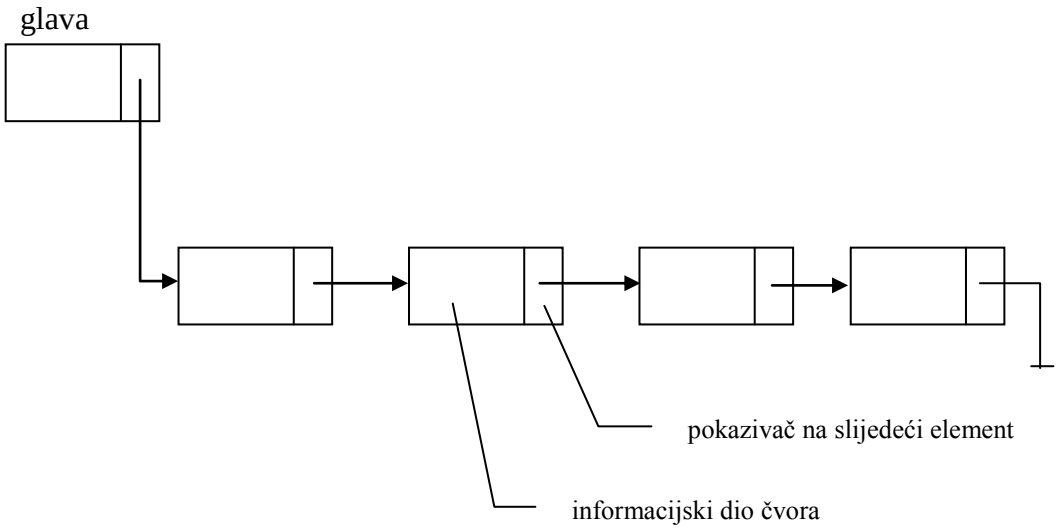
Obrada podataka često uključuje pohranu i obradu podataka organiziranih u linearne liste. Jedan od načina pohrane takvih podataka je pomoću polja. Linearna ovisnost podataka u polju odraz je njihove pozicije u memoriji a ne neke informacije sadržane u samim elementima polja. Zbog toga se može jednostavno izračunati adresa elementa polja. Međutim, upotreba polja za pohranu liste ima i nedostatke : umetanje i brisanje elementa je vremenski zahtjevno. Drugi nedostatk upotrebe polja je taj što unaprijed trebamo znati veličinu polja te se ta veličina ne može mijenjati tokom izvođenja programa.

Drugi način pohrane liste je upotrebom *dinamičke strukture podataka*. Svaki element liste sastoji se od sadržaja elementa i pokazivača koji sadrži adresu slijedečeg elementa liste. Kod tog načina zapisa liste, susjedni elementi liste ne trebaju biti smješeni na susjednim memorijskim adresama. Takav zapis liste omogućava jednostavnije brisanje i umetanje elemenata u listu.

JEDNOSTRUKO POVEZANE LISTE

Jednostruko povezana lista je linearni skup podataka, koji se zovu *čvorovi*, poredanih pomoću *pokazivača*. Svaki čvor podijeljen je u dva dijela : prvi dio sadržava informacije elementa, a drugi dio sadržava adresu slijedečeg čvora u listi. Pokazivač zadnjeg čvora sadržava *null* pokazivač. Lista sadrži i pokazivač liste, koji se obično zove *glava*, koji sadrži adresu prvog člana liste. Posebni slučaj liste je lista bez čvorova. Takva lista se zove *null lista* ili *prazna lista*.

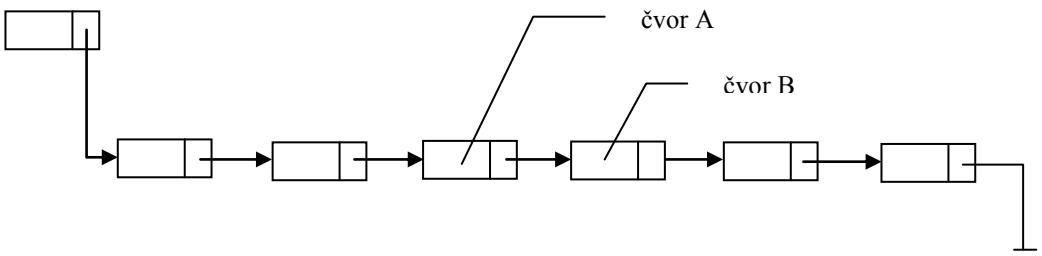
Na slici 3.2. prikazan je shematski dijagram jednostruko povezane liste od 6 čvorova.



Slika 3.2. Jednostruko povezana lista

Umetanje u linearnu listu

Neka je LISTA linearna lista sa susjednim čvorovima A i B, kao što je prikazano na slici 3.3.

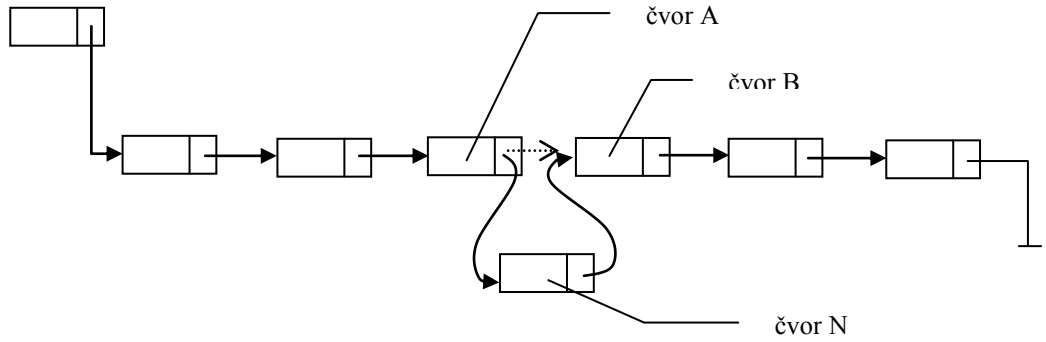


Slika 3.3. Lista prije umetanja čvora N

Da bismo ubacili čvor N u listu između čvorova A i B, trebamo:

a) pokazivač čvora A postaviti na adresu čvora N
b) pokazivač čvora N postaviti na adresu čvora B

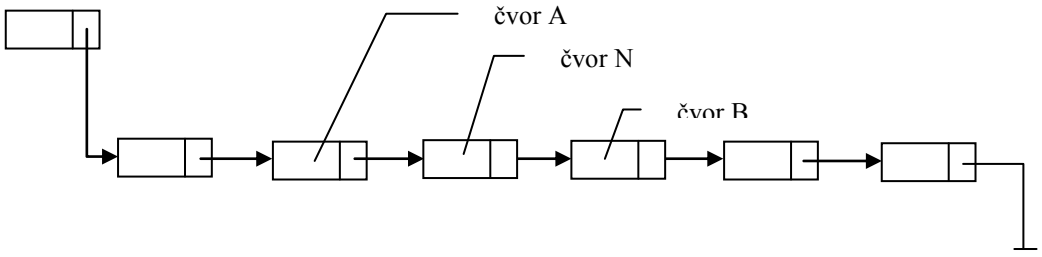
Slika 3.4. ilustrira umetanje čvora N u listu.



Slika 3.4. Lista nakon umetanja čvora N

Brisanje iz linearne liste

Neka je LISTA linearna lista sa čvorom N između čvorova A i B, kao što je prikazano na slici 3.5.

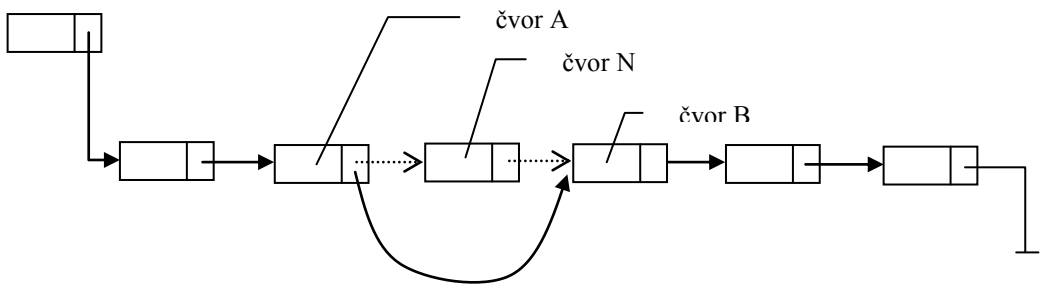


Slika 3.5. Lista prije brisanja čvora N

Da bismo izbacili čvor N iz liste, trebamo:

a) pokazivač čvora A postaviti na adresu čvora B

Na slici 3.6. prikazana je lista nakon brisanja čvora N.



Slika 3.6. Lista nakon izbacivanja elementa N

PRIPREMA ZA VJEŽBU :

Struktura osnovnog elementa se u C-u može definirati ovako :

```
struct OE_ {
 int x;
 struct OE_ *sljedeci;
}
```

U ovome primjeru x je cjelobrojni podatak, međutim, općenito može biti bilo koji tip podataka.
S ovime nastaje novi tip podataka **`struct`** `OE_`. Kako bi ime tog novog tipa podataka bilo još jednostavnije dobra praksa u C-u je napraviti sljedeću naredbu:

```
typedef struct OE_ OE;
```

Nakon ovakve naredbe novi tip podataka se zove jednostavnije: `OE`.

Naposljetku, kako bi mogli koristiti povezani popis, moramo isto ovako globalno definirati glavu (pokazivač na prvi element) popisa. U C-u to se radi naredbom:

```
OE *prvi = NULL;
```

VJEŽBA:

Zadatak 1. Potrebno je generirati slučajni Niz V od N elemenata. Nakon toga iz tog niza formirati povezani popis. Mjeriti i usporediti vrijeme potrebno za kreiranje niza i povezanog popisa. Nakon toga napraviti algoritam sekvencijalnog pretraživanja nad nizom i nad povezanim popisom. Opet mjeriti vrijeme, što je brže?

Zadatak 2 (za veću ocjenu). Pronaći najveći mogući broj elemenata N niza s kojim se može raditi na vašem računalu. Pronaći najveći mogući broj elemenata N povezanog popisa s kojim se može raditi na vašem računalu. O čemu najviše ovisi koliki N može biti?

LABORATORIJSKA VJEŽBA 2 - ALGORITMI I STRUKTURE PODATAKA

Prezime i ime: _____

Broj indeksa: _____

Zadatak 1.

A) Zašto ne možemo provesti algoritam BINARNOG PRETRAŽIVANJA na povezanom popisu?

_____

_____

B) Popuniti tablicu tako da napravite i pokrenete programe (unijeti vrijeme u milisekundama, prilikom testiranja programa za pretraživanje, zadati vrijednost X koje nema u nizu ili povezanom popisu, tako da algoritam traje najduže):

Broj elem. N	**Formiranje niza [ms]**	**Formiranje PP [ms]**	**Pretraživanje niza [ms]**	**Pretraživanje PP [ms]**

C) Skicirajte vrijeme izvršavanja u odnosu na broj elemenata niza na grafu:

D) Napišite vlastiti komentar (što je brže?, vremenska složenost)

_____

_____

_____

Zadatak 2 (za veću ocjenu)

Moje računalo: CPU _____ RAM _____

Najveći mogući niz: _____ elemenata

Najveći mogući povezani popis: _____ elemenata

Objašnjenje:

_____

_____