

## Računske vježbe 8

1. Elementi liste su cijeli brojevi. Sastaviti program koji vrši sumiranje brojeva upisanih u čvorovima liste korišćenjem rekurzivne funkcije.

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

struct str
{
    int a;
    struct str *sled;
};

int sabiranje(struct str *);

int main()
{
    struct str *elem, *preth, *glava;
    int i, n;

    puts("Unijeti broj elemenata liste:");
    scanf("%d", &n);

    puts("Unijeti elemente liste:");
    for(i=0; i<n; i++)
    {
        elem = (struct str *)malloc(sizeof(struct str));
        scanf("%d", &(elem->a));
        elem->sled=NULL;

        if(i==0)
        {
            preth=elem;
            glava=elem;
        }
        else
        {
            preth->sled=elem;
            preth=elem;
        }
    }
    printf("Zbir je %d", sabiranje(glava));
}

int sabiranje(struct str *lista)
{
    if(lista->sled==NULL)
        return lista->a;
    else
        return lista->a + sabiranje(lista->sled);
}
```

2. Elementi liste su sortirani cijeli brojevi u rastućem poretku. Sastaviti program koji učitava cijeli broj i smješta ga u sortiranu listu tako da se ne naruši sortiranost. Umetanje tog broja izvršiti pomoću funkcije, a ne u okviru glavnog programa.

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

struct str
{
    int a;
    struct str *sled;
};

struct str * umetanje(struct str *, int);

int main()
{
    struct str *elem, *preth, *glava, *rez;
    int i, n, broj;

    puts("Unijeti broj elemenata liste");
    scanf("%d", &n);

    puts("Unijeti elemente liste");
    for(i=0;i<n;i++)
    {
        elem=(struct str *)malloc(sizeof(struct str));
        scanf("%d", &(elem->a));
        elem->sled=NULL;
        if(i==0)
        {
            preth=elem;
            glava=elem;
        }
        else
        {
            preth->sled=elem;
            preth=elem;
        }
    }

    printf("Unijeti trazeni broj:");
    scanf("%d", &broj);

    rez=umetanje(glava, broj);

    for(i=0;i<=n;i++)
    {
        printf("%d ", rez->a);
        rez=rez->sled;
    }
}
```

```

struct str * umetanje(struct str *lista, int broj)
{
    struct str *umet, *poc, *pr;

    //u promjenljivoj poc pamtimo glavu liste
    poc=lista;

    umet=(struct str *)malloc(sizeof(struct str));
    umet->a=broj;

//dio programa gdje se vrši provjera da li broj treba umetnuti na početku liste
    if(broj<=lista->a)
    {
        umet->sled=lista;
        return umet;
    }

//dio programa gdje se vrši provjera da li broj treba umetnuti negdje unutar liste
    pr=lista;
    lista=lista->sled;
    while(lista!=NULL)
    {
        if(broj<=lista->a)
        {
            pr->sled=umet;
            umet->sled=lista;
            return poc;
        }
        else
        {
            pr=lista;
            lista=lista->sled;
        }
    }

//ako program stigne do ove linije onda broj treba umetnuti na kraju liste
    pr->sled=umet;
    umet->sled=NULL;
    return poc;
}

```