

# Kružne liste

SPA2

Napisati program koji od  $n$  reči sa ulaza formira i ispisuje kružnu listu.

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

struct lista{
    char *rec;
    struct lista *rep;
};

#define novi(x) x=(struct lista *)malloc(sizeof(struct lista))

void ispis(struct lista *);
void dodaj(struct lista **,char*);
struct lista* form(int);
```

Napisati program koji od  $n$  reči sa ulaza formira i ispisuje kružnu listu.

```
void dodaj(struct lista **p, char *s){
    struct lista *temp, *poc,*tek;
    novi(temp);
    if (!temp) {
        printf("Greska pri alokaciji memorije\n");
        exit(0);}
    temp->rec=(char *)malloc(strlen(s)+1);
    strcpy(temp->rec,s);
    poc=*p;
    if (!poc) {
        temp->rep=temp;      poc=temp;}
    else {
        tek=poc;
        while (tek->rep!=poc) tek=tek->rep;
        temp->rep=tek->rep;  tek->rep=temp;}
    *p=poc;
}
```

Napisati program koji od  $n$  reči sa ulaza formira i ispisuje kružnu listu.

```
struct lista *form(int n){
    int i;
    struct lista *p;
    char s[50];
    p=NULL;
    for(i=0;i<n;i++){
        scanf("%s",s);
        dodaj(&p,s);
    }
    return p;
}
```

Napisati program koji od  $n$  reči sa ulaza formira i ispisuje kružnu listu.

```
void ispis(struct lista *p){
    struct lista *pom;
    if (p) {
        pom=p;
        do {
            printf("%s\t",pom->rec);
            pom=pom->rep;
        } while (pom!=p);
    }
    printf("\n");
}
```

Napisati program koji od  $n$  reči sa ulaza formira i ispisuje kružnu listu.

```
main(){
    int i,n;
    char c;
    struct lista *p;
    p=NULL;
    scanf("%d",&n);
    p=form(n);
    ispis(p);
}
```

# Koliko je elemenata ima u listi?

```
int broj_elem(struct lista *p){
    int n=0;
    struct lista *tek;
    if (p) {
        tek=p;
        do{
            n++;
            tek=tek->rep;
        }while(tek!=p);
    }
    return n;
}
```

Napisati program koji formira kružnu listu celih brojeva tako što se nakon prva dva broja, svaki sledeći dodaje između proizvoljnog para brojeva (prvog para na koji naiđe), ali tako da su oba broja veća od broja koji se dodaje. Sa formiranjem liste stati kada uneti broj nije moguće dodati u listu tako da se zadovolji opisani uslov.

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

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

```
struct lista{  
    int broj;  
    struct lista *rep;  
};
```

```
#define novi(x) x=(struct lista *)malloc(sizeof(struct lista))
```

```
void ispis(struct lista *);  
int dodaj(struct lista **,int);  
struct lista* form();
```



```
int dodaj(struct lista **p, int k){
    struct lista *temp, *poc,*tek;
    int ind=0;
    novi(temp);
    if (!temp) {
        printf("Greska pri alokaciji memorije\n");
        exit(0); }
    temp->broj=k;
    poc=*p;
    if (!poc) {
        temp->rep=temp;
        poc=temp;
        ind=1;
    }
    else if (poc==poc->rep) {
        temp->rep=poc;
        poc->rep=temp;
        ind=1;
    }
}
```

...

...

```
else {
    tek=poc;
    do{
        if ((tek->broj>k)&&(tek->rep->broj>k)) {
            ind=1;
            temp->rep=tek->rep;
            tek->rep=temp;
        }
        else tek=tek->rep;
    }while((!ind)&& (tek!=poc));
}
*p=poc;
return ind;
}
```

```
struct lista *form(){
    int i;
    struct lista *p;
    p=NULL;
    do{
        scanf("%d",&i);
    } while(dodaj(&p,i));
    return p;
}

void ispis(struct lista *p){
    struct lista *pom;
    pom=p;
    do {
        printf("%5d",pom->broj);
        pom=pom->rep;
    } while (pom!=p);
    printf("\n");
}
```

```
main(){  
    struct lista *p;  
    p=NULL;  
    p=form();  
    ispis(p);  
}
```