Dvostruko povezane liste

SPA2

Napisati program koji od **n** celih brojeva formira i ispisuje dvostruko povezanu listu

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

struct lista{
   int broj;
   struct lista *pret,*sled;
};

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

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

Napisati program koji od **n** celih brojeva formira i ispisuje dvostruko povezanu listu

```
void dodaj(struct lista **poc, struct lista **kraj,int k){
    struct lista *temp;
    novi(temp);
    temp->broj=k;
    if (!temp) {
       printf("Greska pri alokaciji memorije\n");
       exit(0);  }
    if (!(*poc)) {
       temp->pret=temp->sled=NULL;
       *poc=temp;   *kraj=temp;
    }
    else {
       temp->pret=NULL;
       temp->sled=*poc;
       (*poc)->pret=temp;
       *poc=temp;
    }
}
```

Napisati program koji od **n** celih brojeva formira i ispisuje dvostruko povezanu listu

```
void form(struct lista **poc,struct lista **kraj,int n){
   int i,k;
   *poc=NULL;
   *kraj=NULL;
   for(i=0;i<n;i++){
      scanf("%d",&k);
      dodaj(poc,kraj,k);
   }
}
```

Napisati program koji od **n** celih brojeva formira i ispisuje dvostruko povezanu listu

```
void ispis_poc(struct lista *p){
   while(p){
      printf("%5d",p->broj);
      p=p->sled;
   }
   printf("\n");
}

void ispis_kraj(struct lista *p){
   while(p){
      printf("%5d",p->broj);
      p=p->pret;
   }
   printf("\n");
}
```

Napisati program koji od **n** celih brojeva formira i ispisuje dvostruko povezanu listu

```
main(){
   int n;
   struct lista *poc,*kraj;
   scanf("%d",&n);
   form(&poc,&kraj,n);
   ispis_poc(poc);
   ispis_kraj(kraj);
}
```

a) Koliko elemenata liste ima svojstvo da se ispred i iza tog elementa nalaze parni brojevi?

```
int parni(struct lista *p){
   int k=0;
   p=p->sled;
   while(p->sled){
      if ((p->pret->broj%2==0) && (p->sled->broj%2==0)) k++;
      p=p->sled;
   }
   return k;
}
```

b) Iz formirane liste izbaciti neparne brojeve.

```
void izbaci(struct lista **poc,struct lista **kraj){
   struct lista *pom,*p;
   pom=*poc;
   while(pom){
      if (pom->broj%2==1){
    p=pom;
    if (pom->sled) pom->sled->pret=pom->pret;
    else *kraj=pom->pret;
    if (pom->pret) pom->pret->sled=pom->sled;
    else *poc=pom->sled;
    pom=pom->sled;
    free(p);
      }
      else pom=pom->sled;
   }
}
```

Stek

Napisati program za računanje vrednosti izraza sa 4 osnovne operacije(+, -, *, /), pri čemu su prisutne sve zagrade, tj. (a op b)

Vrednost izraza

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

struct stek{
   char *sadrzaj;
   struct stek *rep;
};

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

int isoper(char c){
   if((c=='+') || (c=='-') || (c=='*') || (c=='/')) return 1;
   return 0;
}
```

Vrednost izraza

```
struct stek* push(struct stek *p,char *s){
    struct stek *temp;
    novi(temp);
    if(!temp) exit(0);
    temp->sadrzaj=(char *)malloc(strlen(s)+1);
    strcpy(temp->sadrzaj,s);
    temp->rep=p;
    p=temp;
    return p;
}

struct stek* pop(struct stek *p){
   struct stek *temp;
   temp=p;
   p=p->rep;
   free(temp);
   return p;
}
```

Vrednost izraza

```
int izracunaj(){
   char s[50],ch,op;
   int l,d,r,br=0;
   struct stek *poc=NULL;
   while((ch=getchar())!='\n'){
      if (isdigit(ch)){
        while(isdigit(ch)) {
          s[br++]=ch;
          ch=getchar();
        }
        s[br]='\0';
        br=0;
        poc=push(poc,s);
     }
     if(isoper(ch)) {
        s[0]=ch;   s[1]='\0';
        poc=push(poc,s);
     }
...
```

Vrednost izraza

```
...
if(ch==')'){
     d=atoi(poc->sadrzaj);    poc=pop(poc);
     op=poc->sadrzaj[0];      poc=pop(poc);
     l=atoi(poc->sadrzaj);    poc=pop(poc);
     switch (op) {
        case '+': r=l+d; break;
        case '-': r=l-d; break;
        case '*': r=l*d; break;
        case '/': if (d) r=l/d;
                  else {
                     printf("Deljenje nulom nije definisano\n");
                     exit(0);
                  };
                  break;

     }
...
```

Vrednost izraza

```
...
          itoa(r,s,10);
          poc=push(poc,s);
        }
    }
    r=atoi(poc->sadrzaj);
    free(poc);
    return r;
}

main(){
    int a;
    a=izracunaj();
    printf("%d\n",a);
}
```