Operațiile de adăugare/ ștergere/ căutare în LSI

Adăugarea unui nod la listă

1. adăugare la început
2. adăugare la sfârșit
3. adăugare în interior
   - 3.1. după un nod
   - 3.2. înainte de un nod

Căutarea unui nod în listă

4. căutare după valoarea informațională
5. căutare după poziție

Eliminarea unui nod din listă

6. eliminarea primului nod;
7. eliminarea ultimului nod;
8. eliminarea unui nod din interiorul listei.

Adăugarea unui nod la listă

1. adăugare la început

```
void adaugInc(nod*& v, int nr)
{
   nod* c=new nod;
   c->info=nr;
   c->urm=NULL;
   if(v==NULL) v=c;
   else
   {
      c->urm=v;
      v=c;
   }
}
```

Adăugarea unui nod la listă

2.adăugare la sfârșit

```
void adaugSf(nod*& p, int nr)
{
   nod* c=new nod;
   c->info=nr
   c->urm=NULL;
   if(p==NULL) p=c;
   else
   {
     nod *u=p;
     while(u->urm!=NULL) u=u->urm;
     u->urm=c;
   }
}
```

Adăugarea unui nod la listă

3.adăugare în interior

3.1. după un nod

```
void addAfter(nod *q, nod *&ultim)
{
  nod *p=new nod;
  p->info=x;
  p->urm=q->urm;
  q->urm=p;
  if (q==ultim) ultim=p;
}
```

Adăugarea unui nod la listă

a3.adăugare în interior

3.2. înainte de un nod

```
void addBefore(nod *q, nod *&ultim)
{
     nod *p=new nod;
     p->info=q->info;
     q->info=x;
     p->urm=q->urm;
     q->urm=p;
     if (q==ultim) ultim=p;
}
```

Căutarea unui nod în listă

4.căutare după valoarea informațională

```
nod * cautInfo(nod *prim,int nr)
{
   for (nod *p=prim; p!=NULL && p->info!=nr; p=p->urm);
   return p;
}
```

Căutarea unui nod în listă

5.căutare după poziție

```
nod *cautPoz(nod *prim, int poz)
{
    nod *p=prim;
    int nr=1;
    while(p!=NULL && nr<poz)
                  p=p->urm, nr++;
   return p;
}
```

Eliminarea unui nod din listă

6.eliminarea primului nod;

```
void eliminFirst(nod *&prim)
 {
   nod q=prim;
   prim=prim->urm;
   delete q;
}
```

Eliminarea unui nod din listă

7.eliminarea ultimului nod;

```
void eliminLast(nod *prim,nod *&ultim)
{
   nod *p,*q=ultim;
   for (p=prim; p->urm->urm!=NULL; p=p->urm);
   p->urm=NULL;
   ultim=p;
   delete q;
}
```

Eliminarea unui nod din listă

8.eliminarea unui nod din interiorul listei.

```
void eliminIn(nod *p,nod *&ultim)
{
   nod *q=p->urm;
   p->info=p->urm->info;
   p->urm=p->urm->urm;
   delete q;
   if (p->urm==NULL) ultim=p;
}
```