Prof. Dr. Agni Dika
Version punues.
Nuk është për shumëzim.

9

Listat e lidhura

Listat e lidhura

Fushat me të dhëna, p.sh. siç janë vektorët, shpeshë quhen edhe **lista sekuenciale** (ang. sequential list), kjo sepse vendosja e anëtarëve të tyre në memorie është sekuenciale (anëtarët vendosen në lokacione të cilët gjenden njëri pas tjetrit - në sekuencë). Shfrytëzimi i tyre është mjaft praktik gjatë procesit të sortimit të të dhënave sipas madhësisë, ose kërkimit të të dhënave brenda tyre. Por, mes tjerash, e metë e listave sekuenciale është sepse insertimi i anëtarëve të rinjë brenda tyre nuk është i lehtë, ose fshirja e plotë e anëtarëve të tyre është e pamundur.

Një mënyrë tjetër e ruajtjes dhe operimit me të dhëna është ajo e organizimit të tyre në **lista të lidhura** (ang. linked list). Si njësi elementare në strukturën e listave të lidhura është *nyja* (ang. node), në të cilën ruhen të dhënat dhe lidhjet me pjesët tjera të listës.

Te listat e lidhura, për dallim nga fushat (vektoreët, matricat), të dhënat në memorie nuk është e domosdoshme të vendosen njëra pas tjetrës. Kjo, mes tjerash e lehtëson procedurën e shtimit të nyjeve të reja në listë, zhvendosjen e atyre ekzistuese, ose edhe fshirjen e nyjeve brenda listës, pa ndikim në nyjet tjera që përfshihen në listë.

Për fushat, përkatësisht listat sekvenciale të të dhënave mund të thuhet se paraqesin **struktura me madhësi fikse të të dhënave** (ang. fixed-size data structures) [12], sepse madhësia e tyre nuk mund të ndryshohet gjatë ekzekutimit të programit. Kurse, nga ana tjetër, listat e lidhura llogariten si **struktura dinamike të të dhënave** (ang. dynamic data struktures) [12], sepse nyjet (të dhënat) brenda tyre vendosen ose edhe fshihen në mënyrë dinamike, gjë që mundëson rritjen ose zvogëlimin e tyre, përkatësisht ndryshimin e numrit të nyjeve që përfshihen brenda listave. Numri i nyjeve mund të rritet derisa ka memorie të lirë në kompjuter, por sipas nevojës edhe mund të zvogëlohet.

Deklarimi dhe mbushja e nyjes

Nyja (ang. node), si njësi elementare e listës, në rastin e saj më të thjeshtë përmbanë dy komponete, ashtu siç shihet në *Fig.1*.

Fig.1
Nyja në formën e saj elementare

Në komponenten **`data`** të nyjes ruhen të dhëna të tipeve të ndryshme (përfshirë edhe objekte), kurse komponenta **`next`** paraqet një pointer në të cilin ruhet adresa e nyjes vijuese në listë me të cilën ajo lidhet.

Meqë në nyje së paku përfshihen dy komponente, ajo mund të deklarohet si *strukturë* ose *klasë*, gjë që nuk ka dallim. Në pjesën vijuese për këtë qëllim do të shfrytëzohet struktura.

Shembull Programi përmes së cilit definohet dhe mbushet me të dhëna një nyje e listës së lidhur.

```
// Prg1
#include <iostream>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
};

int main()
{
     nyje *n;
     n=new nyje;  // Te pointeri n ruhet adresa e nyjes së re
     n->data=25;
     n->next=NULL;
     cout << "\nMbushja përfundoi";
     cout << endl;
return 0;
}
```

Pas ekzekutimit të programit të dhënë, nyja do të duket ashtu siç është treguar në Fig.

Fig.

Këtu, me rrethin e vizatuar në pjesën e poshtme të nyjes është treguar se vlera brenda kësaj pjese është zero, gjë që e nënkupton moslidhjen e nyjes me një nyje tjetër.
Për krijimin e tabelës vijuese shih komentin më vonë!!!

```
 1  // Prg1
 2  #include <iostream>
 3  using namespace std;
 4  struct nyje
 5  {
 6     int data;
 7     nyje *next;
 8  };
 9
10  int main()
11  {
12     nyje *n;
13     n=new nyje;
14     n->data=25;
15     n->next=NULL;
16     cout << "\nMbushja përfundoi";
17     cout << endl;
18  return 0;
19  }
```

Në rreshtat me numra rendor prej 4 deri në 8 është definuar struktura **`nyje`** me dy komponentet e saj. Komponenta e parë **`data`** i përket pjesës së nyjes ku ruhet informata, kurse komponenta e dytë **`next`** është pointer i tipit **`nyje`**, siç është edhe vetë struktura.

Struktura e cila përmbanë komponente që është pointer i tipit të njëjtë me vetë strukturën quhet *strukturë vetë-referente* (ang. self-referential structure). Duke shfrytëzuar struktura të tilla vetë-referente, mund të krijohen *struktura të lidhura të dhënash* (ang. linked data struktures), siç janë: listat e lidhura, stacks, queues dhe pemët. Nëse për definimin e nyjeve shfrytëzohen klasat, atëherë bëhet fjalë për *klasa vetë-referente* (ang. self-referencial classes).

Në rreshtin 12 të programit është deklaruar pointeri **`n`** i typit nyje, kurse përmes komandës në rreshtin 13:

```
n=new nyje;
```

dikund në memorie deklarohet një *variabël dinamike* (ang. dynamic variable) *e paemëruar*, e cila është e tipit **`nyje`** dhe adresa e saj ruhet te pointeri **`n`**.

Përmes komandës në rreshtin 14 të programit, në pjesën **`data`** të nyjes vendoset vlera 25. Meqë në listë ka vetëm një nyje, përmes komandës

në rreshtin 15, pointeri **next** i saj mbushet me vlerën *zero* (**NULL**), për të treguar se nyja nuk lidhet me nyje tjetër.

Nëse ekzekutohet programi i dhënë, pas mbushjes së nyjes me të dhëna, për të treguar fundin e ekzekutimit të tij, në ekran do të shtypet mesazhi

```
Mbushja përfundoi
```

Shtypja e përmbajtjes së nyjes

Pas mbushjes së nyjes me të dhëna, përmbajtja e saj mundet të lexohet dhe të shtypet duke e shfrytëzuar pointerin në nyje.

Shembull

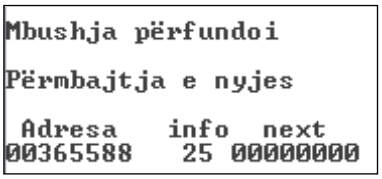
Programi përmes së cilit mbushet me të dhëna një nyje e listës së lidhur dhe pastaj lexohet dhe shtypet përmbajtja e saj.

```
// Prg2
#include <iostream>
#include <iomanip>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
};

int main()
{
     nyje *n;
     n=new nyje;
     n->data=25;
     n->next=NULL;
     cout << "\nMbushja përfundoi";

     cout << "\n\nPërmbajtja e nyjes";
     cout << "\n\n Adresa  Vlera Lidhja\n\n";
     cout << setw(8) << n
          << setw(5) << n->data
          << setw(9) << n->next << endl;
     cout << endl;
```

```
return 0;
}
```

Rez.

Të riekzekutohet!
Fig.



Fillimi dhe fundi i listës

Duke shfrytëzuar pointer përkatës, për çdo listë zakonisht ruhen edhe adresat ku lista fillonë dhe përfundonë.

Shembull

Programi përmes së cilit definohet dhe mbushet me të dhëna një nyje e listës së lidhur. Listës i shtohen edhe pointerët për fillimin dhe fundin e saj.

```
// Prg3
#include <iostream>
#include <iomanip>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
};

int main()
{
     nyje *n,*first=NULL,*last=NULL;
     n=new nyje;
     n->data=25;
     n->next=NULL;
     first=n;
     last=n;
     cout << "\nMbushja përfundoi";
```

```
     cout << "\n\nFillimi: " << first
          << "   Fundi: " << last;
     cout << "\n\nPërmbajtja e nyjes";
     cout << "\n\n Adresa  Vlera Lidhja\n\n";
     cout << setw(8) << n
          << setw(5) << n->data
          << setw(9) << n->next << endl;
     cout << endl;
return 0;
}
```

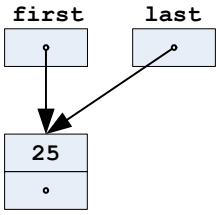
Rez.

Të riekzekutohet!

```
Mbushja përfundoi

Fillimi: 00365588   Fundi: 00365588

Përmbajtja e nyjes

 Adresa   info  next
00365588    25 00000000
```

Fig.

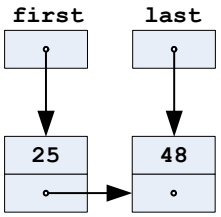
Fig.



Mbushja e dy nyjeve të listës

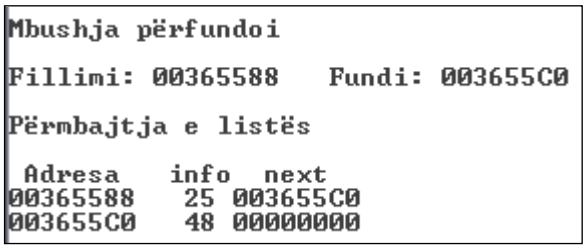
```
// Prgxx
#include <iostream>
#include <iomanip>
using namespace std;

struct nyje
{
     int data;
     nyje *next;
};

int main()
```

```
{
     nyje *n,*first=NULL,*last=NULL;

     n=new nyje;
     n->data=25;
     n->next=NULL;
     first=n;     // Pointer i tipit nyje
     last=n;      // Pointer i tipit nyje

     n=new nyje;
     last->next=n;
     n->data=48;
     n->next=NULL;
     last=n;
     cout << "\nMbushja përfundoi";
// ------------------------------------------------------------
     cout << "\n\nFillimi: " << first
          << "   Fundi: " << last;

     cout << "\n\nPërmbajtja e listës";
     n=first;
     cout << "\n\n Adresa  Vlera Lidhja\n\n";
          cout << setw(8) << n
               << setw(5) << n->data
               << setw(9) << n->next << endl;
     n=n->next;  // Te n ruhet adresa që gjendet te next
          cout << setw(8) << n
               << setw(5) << n->data
               << setw(9) << n->next << endl;
     cout << endl;
return 0;
}
```



Këtu, pointerët **`first`** dhe **`last`** janë të tipit **`nyje`**, por pjesët **`data`** të tyre nuk i shfrytëzojmë.

Tjetër info me data!!



Pjesa e fundit e programit për shtypje të listës mund të duket edhe kështu:

```
     ..............................
     cout << "\n\nPërmbajtja e listës";
     n=first;
     cout << "\n\n Adresa  Vlera Lidhja\n\n";
          cout << setw(8) << n
               << setw(5) << n->data
               << setw(9) << n->next << endl;
          cout << setw(8) << n->next
               << setw(5) << n->next->data
               << setw(9) << n->next->next << endl;
     cout << endl;
return 0;
}
```

Në praktikë, për shtypje të përmbajtjes së listës, pavarsisht nga numri i nyjeve të saj, shfrytëzohet një unazë e mbyllur. Në vijim është dhënë pjesa e fundit e programit paraprak, ku përfshihet unaza në fjalë.

```
     ..................................
     cout << "\n\nPërmbajtja e listës";
     n=first;
     cout << "\n\n Adresa   data  next\n";
     while (n!=0)
     {
          cout << setw(8) << n
               << setw(5) << n->data
               << setw(9) << n->next << endl;
          n=n->next;  // Te n ruhet adresa që është te next
     }
     cout << endl;
return 0;
}
```

Shfrytëzimi i një funksioni për shtypje

```
// Prgxxy
#include <iostream>
#include <iomanip>
using namespace std;

struct nyje
{
     int data;
     nyje *next;
};

void PrintList(nyje *first);

int main()
{
     nyje *n,*first=NULL,*last=NULL;

     n=new nyje;
     n->data=25;
     n->next=NULL;
     first=n;
     last=n;

     n=new nyje;
     last->next=n;
     n->data=48;
     n->next=NULL;
     last=n;
     cout << "\nMbushja përfundoi";

     cout << "\n\nFillimi: " << first
          << "   Fundi: " << last;
     PrintList(first);
return 0;
}

void PrintList(nyje *first)
{
     nyje *n=first;
     cout << "\n\nPërmbajtja e listës";
     cout << "\n\n Adresa   data  next\n";
     while (n!=0)
     {
          cout << setw(8) << n
```

```
                << setw(5) << n->data
                << setw(9) << n->next << endl;
            n=n->next;
      }
      cout << endl;
}
```

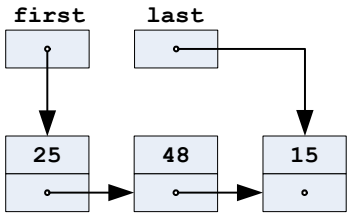
Mbushja e listës përmes unazës variabile

```
// PrgxxxV
#include <iostream>
#include <iomanip>
using namespace std;

struct nyje
{
      int data;
      nyje *next;
};

void PrintList(nyje *first);

int main()
{
      nyje *n,*first=NULL,*last=NULL;
      int x,i=1;
      char z='P';
      while (z=='P')
      {
            n=new nyje;    // Te pointeri n ruhet adresa e nyjes së re
            cout << "\nVlera në nyjen e " << i << ": ";
            cin >> x;
            n->data=x;
            n->next=NULL;
            if (first==NULL)
                  first=n;
            else
                  last->next=n;
            last=n;
            cout << "\nNyje tjeter, P-Po, J-Jo: ";
            cin >> z;
            i++;
      }
      cout << "\nMbushja përfundoi";
```

```
     cout << "\n\nFillimi="
          << first
          << "   Fundi="
          << last;
     PrintList(first);
return 0;
}
```

i - numërues i nyjeve të listës. Nuk është i domosdoshëm.
z - variabël e tipit karakter

P.sh.:



Mbushja me të dhëna dhe në fund edhe shtypja e listës rrjedhë kështu:

```
Vlera në nyjen e 1: 25

Nyje tjeter, P-Po, J-Jo: P

Vlera në nyjen e 2: 48

Nyje tjeter, P-Po, J-Jo: P

Vlera në nyjen e 3: 15

Nyje tjeter, P-Po, J-Jo: J

Mbushja përfundoi

Fillimi=003655E8   Fundi=003657A0

Përmbajtja e listës

 Adresa  Vlera Lidhja

003655E8   25 00365710
00365710   48 003657A0
003657A0   15 00000000
```

Duhet tjetër!

Vlerat në nyjen e dytë dhe tretë mund të merren edhe përmes shprehjeve:

```
n->next->next->next
n->next->next->data
```

Mbushja e listës përmes unazës fikse

Këtu, në fillim të mbushjes së listës kompjuterit si vlerë hyrëse i ipet numri i nyjeve **k**.

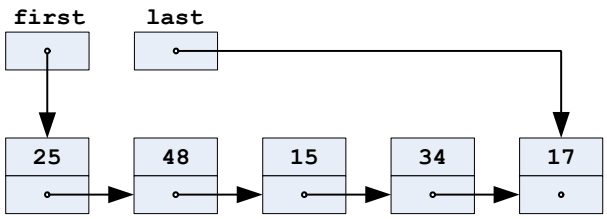
```
// Prgxxy1
#include <iostream>
#include <iomanip>
using namespace std;

struct nyje
{
     int data;
     nyje *next;
};

void PrintList(nyje *first);

int main()
{
     nyje *n,*first=NULL,*last=NULL;
     int x,k,i;
     cout << "\nNumri i nyjeve: ";
     cin >> k;
     for (i=1;i<=k;i++)
     {
          n=new nyje;
          cout << "\nVlera në nyjen e " << i << ": ";
          cin >> x;
          n->data=x;
          n->next=NULL;
          if (first==NULL)
               first=n;
          else
               last->next=n;
```

```
            last=n;
      }
      cout << "\nMbushja përfundoi";
      cout << "\n\nFillimi: " << first
            << "   Fundi: " << last;
      PrintList(first);
return 0;
}
```

Shembullin e listës me pesë nyje, procesi i mbushjes së listës dhe në fund edhe shtypja e saj duket kështu:

Tjetër!

```
Numri i nyjeve: 5

Vlera në nyjen e 1: 25

Vlera në nyjen e 2: 48

Vlera në nyjen e 3: 15

Vlera në nyjen e 4: 34

Vlera në nyjen e 5: 17

Mbushja përfundoi

Fillimi: 00365660   Fundi: 00365810

Përmbajtja e listës

 Adresa  Vlera Lidhja

00365660   25 00365768
00365768   48 003657A0
003657A0   15 003657D8
003657D8   34 00365810
00365810   17 00000000
```



Në pjesën vijuese do të shfrytëzohet vetëm mbushja e listës me unazë fikse.

Shfrytëzimi i një funkcioni për mbushje të listës

Për mbushje të listës mund të shfrytëzohet një funksion. Në vijim është treguar dukja e funksionit **load** te i cili për mbushje shfrytëzohet një unazë fikse.

```
// Prgyyyy
#include <iostream>
#include <iomanip>
using namespace std;

struct nyje
{
     int data;
     nyje *next;
};

nyje *load();
void PrintList(nyje *first);

int main()
{
     nyje *first;
     first=load();
     PrintList(first);
return 0;
}

void PrintList(nyje *first)
{
.....................
}

nyje *load()
{
     nyje *n,*first=NULL,*last=NULL;
     int x,k,i;
     cout << "\nNumri i nyjeve: ";
     cin >> k;
     for (i=1;i<=k;i++)
```

```
	{
		n=new nyje;
		cout << "\nVlera në nyjen e " << i << ": ";
		cin >> x;
		n->data=x;
		n->next=NULL;
		if (first==NULL)
			first=n;
		else
			last->next=n;
		last=n;
	}
	cout << "\nMbushja përfundoi";
	cout << "\n\nFillimi: " << first
		<< "   Fundi: " << last;
	return first;
}
```

Mbushja e listës prej fundit

```
// Programi - lista e lidhur e ndertuar nga fundi
#include <iostream>
#include <iomanip>
using namespace std;
struct nyje
{
	int data;
	nyje *next;
};

nyje *loadBackward();
void PrintList(nyje *first);

int main()
{
	nyje *first;
	first=load();
	PrintList(first);
	return 0;
}

nyje *loadBackward()		// Prej fundit
{
	nyje *n,*first=NULL,*last;
	int x,k,i;
```

```
      cout << "\nNumri i nyjeve: ";
      cin >> k;
      for (i=1;i<=k;i++)
      {
            n=new nyje;
            cout << "\nVlera ne nyje: ";
            cin >> x;
            n->data=x;
            n->next=first;
            if (first==NULL)
                  last=n;
            first=n;
      }
      cout << "\nMbushja përfundoi";
      cout << "\n\nFillimi: " << first
            << "   Fundi: " << last;
      return first;
}

void PrintList(nyje *first)
{
      nyje *n=first;
      cout << "\n\nPërmbajtja e listës";
      cout << "\n\n Adresa   data  next\n";
      while (n!=0)
      {
            cout << setw(8) << n
                  << setw(5) << n->data
                  << setw(9) << n->next << endl;
            n=n->next;
      }
      cout << endl;
}
```

Këtu, funksioni **`print`** nuk duhet sepse është dhënë më herët. Tash le të mbetet që ta kemë programin komplet.

Anëtarët e fushës në listë të lidhur

Për arsye praktike, anëtarët e vektorëve, matricave ose edhe fushave shumëdimensionale mund të vendosen në një listë të lidhur. Gjatë kësaj, mund të vendosen të gjithë anëtarët e fushës, ose vetëm anëtarët e caktuar të saj duke i përzgjedhur përmes kushtit përkatës.

Anëtarët e vektorëve në listë

Për vendosje të të gjithë anëtarëve të vektorit në listë të lidhur, mund të shfrytëzohet struktura e programit .... i cili është dhënë më sipër. Kurse për vendosje të anëtarëve të caktuar të tij, para vendosjes duhet të shfrytëzohet kushti përkatës.

Shembull

Vendosja e vetëm anëtarëve pozitiv të vektorit A(k) në një listë të lidhur.
Në vektor të përfshihen anëtarë të njëjtë si edhe te lista paraprake.

```
// PrgVektor1
#include <iostream>
#include <iomanip>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
};

void PrintList(nyje *first);

int main()
{
     nyje *n,*first=NULL,*last=NULL;
     int i;
     const int k=7;
     int A[k]={34,-22,-15,66,11,-72,52};
     for (i=0;i<k;i++)
          if (A[i]>=0)
          {
               n=new nyje;
               n->data=A[i];
               n->next=NULL;
               if (first==NULL)
                    first=n;
               else
                    last->next=n;
```

```
                last=n;
            }
      cout << "\nMbushja përfundoi";
      cout << "\n\nfirst: " << first
           << " last: " << last;
      PrintList(first);
return 0;
}

void PrintList(nyje *first)
{
     ..................
}
```

Rez.

```
Mbushja përfundoi

first: 003655E8 last: 00365690

Përmbajtja e listës

 adress    data  next
003655E8    34 00365620
00365620    66 00365658
00365658    11 00365690
00365690    52 00000000
```

Funksioni print të mos jepet në fund (është i njëjtë me atë që është dhënë më parë).

Nëse duam që të gjithë anëtarët e vektorit të përfshihen në listë, në programin e mësipërm duhet të fshihet komanda `if` përkatëse.

Anëtarët e matricave në listë

Ngjashëm si edhe për vektorët, edhe anëtarët e matricave ose vetëm anëtarë të caktuar të tyre mund të vendosen në një listë të lidhur.

Shembull

Vendosja në listë e anëtarëve të matricës **`A(k,m)`**, të cilët janë më të mëdhenjë se 8 dhe më të vegjël se 22.

```
// PrgMatrica1
#include <iostream>
```

```
#include <iomanip>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
};

void PrintList(nyje *first);

int main()
{
     nyje *n,*first=NULL,*last=NULL;
     int i,j;
     const int k=3,m=5;
     int A[k][m]={{4,1,7,12},
     {9,6,18,23},
     {8,5,34,-7}};
     for (i=0;i<k;i++)
          for (j=0;j<m;j++)
               if (A[i][j]>8 && A[i][j]<22)
               {
                    n=new nyje;
                    n->data=A[i][j];
                    n->next=NULL;
                    if (first==NULL)
                         first=n;
                    else
                         last->next=n;
                    last=n;
               }
     cout << "\nMbushja përfundoi";
     cout << "\n\nfirst: " << first
          << " last: " << last;
     PrintList(first);
return 0;
}

void PrintList(nyje *first)
{
   ..................
}
```

Rez.

```
Mbushja përfundoi

first: 003655E8 last: 00365658

Përmbajtja e listës

 adress    data  next
003655E8    12 00365620
00365620     9 00365658
00365658    18 00000000
```

Pas mbushjes, listës mund t'i shtohen edhe vlera tjera, dhe të gjitha manipulimet tjera me nyjet e listës.

Numërimi i nyjeve të listës

Për t'i numëruar anëtarët e listës përmes një numratori, përkatësisht për gjetje të gjatësisë të listës, duhet të kalohet nëpër të gjitha nyjet e saj

Shembull

Numërimi i nyjeve në listën e cila mbushet me anëtarët negativ të vektorit të dhënë **`A(k)`**.

```
// PrgVektor2
#include <iostream>
#include <iomanip>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
};

int count(nyje *first);

int main()
{
     nyje *n,*first=NULL,*last=NULL;
     int i;
     const int k=7;
     int A[k]={34,-22,-15,66,11,-72,52};
     for (i=0;i<k;i++)
          if (A[i]>=0)
```

```
          {
               n=new nyje;
               n->data=A[i];
               n->next=NULL;
               if (first==NULL)
                    first=n;
               else
                    last->next=n;
               last=n;
          }
     cout << "\nMbushja përfundoi";
     cout << "\nNumri i nyjeve në listë: "
          << count(first)
          << endl;
return 0;
}

int count(nyje *first)
{
     nyje *n=first;
     int k=0;
     while (n!=0)
     {
          k++;
          n=n->next;
     }
     return k;
}
```

Rez.

```
Mbushja përfundoi
Numri i nyjeve në listë: 4
```

Gjetja e anëtarit të caktuar në listë

Qasja në nyjet e listës bëhet vetëm në rrugë sekuenciale. Prandaj kërkimi i vlerës së dëshiruar nëpër të gjitha nyjet e listës duhet të filloi prej nyjes së parë ose të fundit të saj.

Shembull

Mbushja e listës me vlerat e vektorit A(k) të cilët janë më të mëdhenjë se 15. Pastaj gjetja e numrit rendor të nyjes në të cilën gjendet vlera x të cilën vlerë kompjuterit ia japim si vlerë hyrëse përmes tastierës.

```
// Gjetja e anetarit me vlere te caktuar x
#include <iostream>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
};

int main()
{
     nyje *first=NULL, *last=NULL, *n;
     const int k=8;
     int A[k]={55,12,-72,216,-33,9,17,48};
     int i;
     for (i=0;i<k;i++)
          if (A[i]>15)
          {
               n=new nyje;
               n->data=A[i];
               n->next=NULL;
               if (first==NULL)
                    first=n;
               else
                    last->next=n;
               last=n;
          }

// Gjetja
     int x;
     cout << "\nVlera që kërkohet: ";
     cin >> x;
     n=first;
     i=0;    // Numrimi se cili anetar eshte me rradhe ne
liste
     while (n != 0)
     {
          i++;
          if(x==n->data)
          {
               cout << "\nVlera e kërkuar gjendet te nyja e "
```

```
                    << i
                    << ".\n";
               goto end;
          }
          else
               n=n->next;
     }
     cout << "\nVlera e kërkuar nuk u gjet në listë";
end:
     cout << endl;
return 0;
}
```

P.sh. nëse ia japim hyrëse x=17, rez.:

Gjetja e nyjeve te të cilat ruhen vlera të caktuara

Shemull

Mbushja me vlera të vektorit. Gjetja e vlerave me vlerë absolute mes 5 dhe 28.

```
// Gjetja e anetarit me vlere te caktuar x
#include <iostream>
#include <iomanip>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
};

int main()
{
     nyje *first=NULL, *last=NULL, *n;
     const int k=8;
     int A[k]={55,12,-72,216,-33,9,17,48};
     int i,x;
     for (i=0;i<k;i++)
     {
          n=new nyje;
          n->data=A[i];
          n->next=NULL;
```

```
          if (first==NULL)
               first=n;
          else
               last->next=n;
          last=n;
     }
// Gjetja
     i=0;  // Numrimi se cili anetar eshte me rradhe ne liste
     cout << "\n\nVlerat e kërkuara";
     cout << "\n\n Adresa   data  next    nr\n";
     n=first;
     while (n!=0)
     {
          i++;
          x=abs(n->data);
          if(x>3 && x<28)
               cout << setw(8) << n
                    << setw(5) << n->data
                    << setw(9) << n->next
                     << setw(4) << i << endl;
          n=n->next;
     }
     if (i==0)
          cout << "\nVlera e kërkuar nuk u gjet në listë";
     cout << endl;
return 0;
}
```

```
Vlerat e kërkuara

 Adresa   data  next    nr
003655C0   12 003655F8   2
003656A0    9 003656D8   6
003656D8   17 00365710   7
```

Fshirja e listës së lidhur

Me qëllim të lirimit të hapësirës memoruese të cilën e okupon lista e lidhur ajo duhet të fshihet.

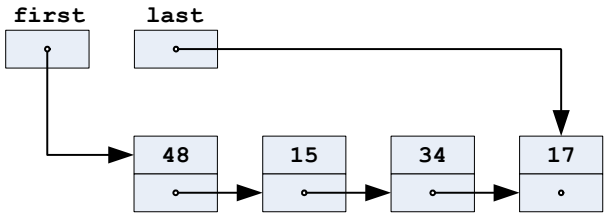
Në fund të thirret funksioni count për të numruar se sa nyje kanë mbetur.

```
// Mbushja dhe fshirja e listes
#include <iostream>
#include <iomanip>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
};
void PrintList(nyje *first);
int main()
{
     nyje *first=NULL, *last=NULL, *n;
     const int k=5;
     int A[k]={7,3,4,9,2},i;
     for (i=0;i<k;i++)
     {
          n=new nyje;
          n->data=A[i];
          n->next=NULL;
          if (first==NULL)
               first=n;
          else
               last->next=n;
          last=n;
     }
// Shtypja
     PrintList(first);
// Fshirja
     cout << "Fillimi i fshirjes së listës";
     cout << "\nfirst=" << first;
     while (first!=NULL)
     {
          n=first;
          first=first->next;
          cout << "\nfirst=" << first;
          delete n;
     }
     last=first;
     cout << "\nlast=" << last;
     cout << "\nFshirja e listës përfundoi\n";
return 0;
}

void PrintList(nyje *first)
{
     ......................
```

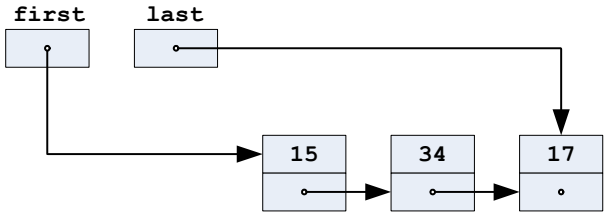
```
}
```

Procedura e fshirjes fillonë me nyjen e parë.
Fillimisht, si në Fig. që është dhënë në fillim:

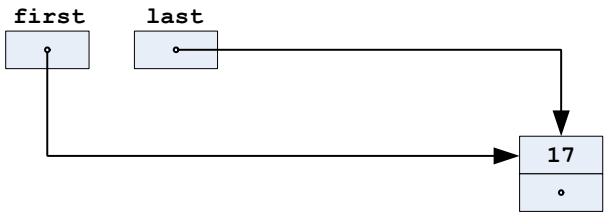
Pas fshirjes së nyjes së parë kemi:



Pas fshirjes së nyjes së dytë lista duket kështu:



Në një moment kur mbetet vetëm nyja e fundit:



first dhe last tregojnë në një nyje.

Mbushja me vlera dhe pastaj fshirja e nyjeve rrjedhë kështu:

```
Përmbajtja e listës

 Adresa    data  next
00365588     7 003655C0
003655C0     3 003655F8
003655F8     4 00365630
00365630     9 00365668
00365668     2 00000000

Fillimi i fshirjes së listës
first=00365588
first=003655C0
first=003655F8
first=00365630
first=00365668
first=00000000
last=00000000
Fshirja e listës përfundoi
```

Ide: si të realizohet fshirja nëse fillohet prej fundit të listës.

Insertimi i nyjes së re

Nyja e re mund të insertohet pas një nyje të zgjedhur.

```
// Insertimi i nyjes
#include <iostream>
#include <iomanip>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
};
void PrintList(nyje *first);
int main()
{
     nyje *first=NULL,*last=NULL,*n;
     const int k=5;
     int A[k]={7,3,4,9,2},i;
     for (i=0;i<k;i++)
     {
          n=new nyje;
          n->data=A[i];
```

```
          n->next=NULL;
          if (first==NULL)
               first=n;
          else
               last->next=n;
          last=n;
     }
// Shtypja
     PrintList(first);

// Insertimi i nyjes
     int p,x,nr;
     nyje *m;
lexo:
     cout << "\nPas cilës nyje insertohet nyja e re: ";
     cin >> p;      // Nyja inserohet pas nyjes së p-të
     if (p<1 || p>k)
     {
          cout << "\nGabim numri i nyjes";
          goto lexo;
     }
     cout << "\nInformata që ruhet në nyje: ";
     cin >> x;
     m=new nyje;
     m->data=x;
// Leximi i listes dhe lidhja e nyjes së re
     n=first;
     nr=1;    // Numratori i nyjeve
     while (n!=0)
     {
          if (p==nr)       // Insertimi i nyjes
          {
               m->next=n->next;
               n->next=m;
               PrintList(first);
               goto jasht;
          }
          n=n->next;
          nr=nr+1;
     }
jasht:
     cout << endl;
return 0;
}

void PrintList(nyje *first)
{
```

```
     ....................
}
```

Këtu duhet vizatimi i komplet listës para shtimit të nyjes si dhe nyja që shtohet - e pa lidhur. Pastaj duhet të vizatohet lista me nyjen e shtuar (por posht - jashtë rendit) dhe lidhjet përkatëse.

Insertimi pas nyjes me vlerë të caktuar

Në këtë rast nuk pyesim se a kemi mbërri te nyja e nr-të por se a është informata në nyje me vlerë të barabartë me vlerën të cilën e kemi dhënë si vlerë hyrëse.

```
// Insertimi i nyjes pas vleres se caktuar
#include <iostream>
#include <iomanip>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
};
void PrintList(nyje *first);
int main()
{
     nyje *first=NULL,*last=NULL,*n;
     const int k=5;
     int A[k]={7,3,4,9,2},i;
     for (i=0;i<k;i++)
     {
          n=new nyje;
          n->data=A[i];
          n->next=NULL;
          if (first==NULL)
               first=n;
          else
               last->next=n;
          last=n;
     }
// Shtypja
     PrintList(first);
// Insertimi i nyjes
     int x,nr;
```

```
     nyje *m;
     cout << "\nInformata që ruhet në nyje: ";
     cin >> x;
     m=new nyje;
     m->data=x;
// Leximi i listes dhe lidhja e nyjes së re
     n=first;
     nr=1;     // Numratori i nyjeve
     while (n!=0)
     {
          if (x>n->data)             // Insertimi i nyjes
          {
               m->next=n->next;
               n->next=m;
               PrintList(first);
               goto jasht;
          }
          n=n->next;
          nr=nr+1;
     }
     cout << "\nNyja nuk mund të insertohet";
jasht:
     cout << endl;
return 0;
}

void PrintList(nyje *first)
{
     ......................
}
```

Vizatimi i listës!!

Insertimi në fillim të listës

Insertimi në fillim të listës.

```
// Insertimi i nyjes
#include <iostream>
#include <iomanip>
using namespace std;
struct nyje
{
     int data;
     nyje *next;
```

```
};
void PrintList(nyje *first);
int main()
{
      nyje *first=NULL,*last=NULL,*n;
      const int k=5;
      int A[k]={7,3,4,9,2},i;
      for (i=0;i<k;i++)
      {
            n=new nyje;
            n->data=A[i];
            n->next=NULL;
            if (first==NULL)
                  first=n;
            else
                  last->next=n;
            last=n;
      }
// Shtypja
      PrintList(first);

// Insertimi i nyjes
      int x;
      nyje *m;
      cout << "\nInformata që ruhet në nyje: ";
      cin >> x;
      m=new nyje;
      m->data=x;
// Lidhja e nyjes së re
      n=first;
      m->next=n;
      first=m;
      PrintList(first);
      cout << endl;
return 0;
}

void PrintList(nyje *first)
{
      ......................
}
```

Pjesa me ngjyrë të verdhë përsëritet. Të gjendet një zgjidhje, p.sh., me thirrjen e një funksioni loadList, edhe për shembujt tjerë.

Disa të dhëna në një nyje

Dy të dhëna në një nyje.
Programi i parregulluar.

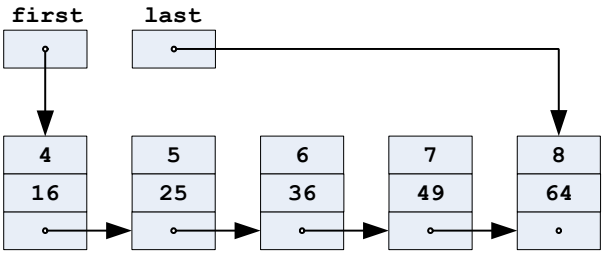
```
// Prgxxya
#include <iostream>
#include <iomanip>
using namespace std;

struct nyje
{
     float a,s;
     nyje *next;
};

int main()
{
     nyje *n,*first=NULL,*last=NULL;
     int i,k;
     float a,s;
     cout << "\nNumri i nyjeve: ";
     cin >> k;
     a=2;
     for (i=1;i<=k;i++)
     {
          s=a*a;
          n=new nyje;
          n->a=a;
          n->s=s;
          n->next=NULL;
          if (first==NULL)
               first=n;
          else
               last->next=n;
          last=n;
          a=a+3;
     }
     cout << "\nMbushja përfundoi";
     cout << "\n\nFillimi: " << first
          << "   Fundi: " << last;
     cout << "\n\nPërmbajtja e listës";
     n=first;
     cout << "\n\n Adresa   r       s      next\n";
     while (n!=0)
```

```
      {
            cout << setw(8) << n
                 << setw(5) << n->a
                  << setw(10) << n->s
                 << setw(9) << n->next << endl;
            n=n->next;
      }
      cout << endl;
return 0;
}
```

Rez.

```
Numri i nyjeve: 5

Mbushja përfundoi

Fillimi: 00365600   Fundi: 00365848

Përmbajtja e listës

 Adresa    r         s        next
00365600     4         16 00365638
00365638     5         25 00365670
00365670     6         36 00365810
00365810     7         49 00365848
00365848     8         64 00000000
```

Lista:



Të merret edhe shembulli kur në një nyje vendosen sipërfaqja dhe perimetri i katrorit.

Gjetja e anëtarit maksimal:

Versioni A

```
// Programi Gjetja e antarit maksimal
// Këtu ruhet pointeri ku gjendet anëtari maksimal
#include <iostream>
#include <iomanip>
using namespace std;

struct Nyje
{
     int a;
     Nyje *p;
};

int main()
{
     int const k=5;
     int i;
     int const D[k]={50,300,150,200,250};
     Nyje n[k];
     Nyje *fillimi,*vijuese;
     fillimi= &n[0];
     cout << "\nAdresa e fillimit të listës: "
          << fillimi
          << "\n";
     for (i=0;i<k;i++)
     {
          n[i].a=D[i];
          if(i!=(k-1))
               n[i].p=&n[i+1];
          else
               n[i].p=NULL;
     }

// Shtypja
     vijuese=fillimi;
     cout << "\n  Vlera    Adresa vijuese\n\n";
     while (vijuese != 0)
     {
          cout << setw(6)
               << vijuese->a
               << "      "
                << setw(10)
               << vijuese->p
```

```
                    << "\n";
               vijuese=vijuese->p;
          }

// Vlera maksimale ruhet te pointeri max
          Nyje *max;
          max=fillimi;
          vijuese=&n[1];
          while (vijuese != 0)
          {
               if (vijuese->a > max->a)
                    max->a=vijuese->a;
               else
                    vijuese=vijuese->p;
          }
          cout << "\nVlera maksimale: "
               << max->a
               << "\n";
return 0;
}
```

Shembull

```
// Gjetja e shumës
#include <iostream>
#include <iomanip>
using namespace std;

struct Nyje
{
     int a;
     Nyje *p;
};

int main()
{
     int const k=5;
     int i;
     int const D[k]={50,300,150,200,250};
     Nyje n[k];
     Nyje *fillimi,*vijuese;
     fillimi= &n[0];
     cout << "\nAdresa e fillimit të listës: "
          << fillimi
          << "\n";
     for (i=0;i<k;i++)
```

```
      {
            n[i].a=D[i];
            if(i!=(k-1))
                  n[i].p=&n[i+1];
            else
                  n[i].p=NULL;
      }
// Shtypja
      vijuese=fillimi;
      cout << "\n  Vlera    Adresa vijuese\n\n";
      while (vijuese != 0)
      {
            cout << setw(6)
                  << vijuese->a
                  << "       "
                   << setw(10)
                  << vijuese->p
                   << "\n";
            vijuese=vijuese->p;
      }

// Llogaritja e shumës
      int s=0;
      vijuese=fillimi;
      while (vijuese != 0)
      {
            s=s+vijuese->a;
            vijuese=vijuese->p;
      }
      cout << "\nShuma: "
            << s
            << "\n";
return 0;
}
============================================================

Shembull

// Shtimi i nyjes se re
#include <iostream>
#include <iomanip>
using namespace std;

struct Nyje
{
      int a;
      Nyje *p;
```

```
};

int main()
{
     Nyje n1,n2;
     Nyje *fillimi,*vijuese;
     fillimi= &n1;
     cout << "\nAdresa e fillimit të listës: "
          << fillimi
          << "\n";
     n1.a=50;
     n1.p=&n2;
     n2.a=100;
     n2.p=NULL;
     vijuese=fillimi;
     cout << "\n  Vlera    Adresa vijuese\n\n";
     while (vijuese != 0)
     {
          cout << setw(6)
               << vijuese->a
               << "      "
                << setw(10)
               << vijuese->p
                << "\n";
          vijuese=vijuese->p;
     }

// Shtimi i nyjes se re
     Nyje *n3;
     n3=new Nyje;   // Nyje e vetmuar dikund jashte listes
     n2.p=n3;
     n3->a=49;
     n3->p=NULL;
     vijuese=fillimi;
     cout << "\nPamja e listës pas shtimit"
          << "\n  Vlera    Adresa vijuese\n\n";
     while (vijuese != 0)
     {
          cout << setw(6)
               << vijuese->a
               << "      "
                << setw(10)
               << vijuese->p
                << "\n";
          vijuese=vijuese->p;
     }
     cout << "\n";
```

```
return 0;
}
```