```
#include <iostream>
#include <string>
using namespace std;

struct Nodo {
 string nome;
 Nodo* pSucc;
};
 Nodo* pTesta;        //--- puntatore alla testa della lista
 Nodo* pNew;          //--- puntatore a nuovo Nodo in inserimento
 Nodo*  pTemp;        //--- puntatori temporanei per fIns, fCan, fElim
 Nodo*  pPrec;
 string nomeInp;

void fIns();   void fCan();   //--- prototipi delle funzioni
void fVis();   void fElim();

int main(){
  char sce;
    pTesta = NULL;              //--- inizializzazione lista vuota

    cout<< "\n---Gestione INS/CAN/VIS Lista Nomi---\n";
    cout<< "\n\nscegliere tra :";
    cout<<   "\t i - inserisci  nome\n";
    cout<< "\t\t c - cancella   nome\n";
    cout<< "\t\t v - visualizza nomi\n";
    cout<< "\t\t e - elimina   lista\n";
    cout<< "\t\t u - USCITA\n";
    cout<< "\ndigitare scelta:\n";
    do {
       cin >> sce;
       switch (sce) {
         case 'i':
                    { cin>>nomeInp; fIns();  break; }
         case 'c':
                    { cin>>nomeInp; fCan();  break; }
         case 'v':
                    {                fVis();  break; }
         case 'e':
                    {                fElim(); break; }
         case 'u':
                    return 0;
       }
    } while (1);      //--- ciclo infinito; si esce digitando  u
}
```

Il programma crea dinamicamente in Memoria Centrale una **Lista** di nomi ordinata alfabeticamente.
Scelta la **visualizzazione** il programma segnala che la **Lista è vuota**; per l' **inserimento** dei nomi *luca, anna, zoe, ugo* e *aldo* vengono esposti gli **indirizzi assegnati ai rispettivi Nodi**:

```
Prompt dei comandi
>>listaNomi.exe

---Gestione INS/CAN/VIS Lista Nomi---

scegliere tra :  i - inserisci  nome
                 c - cancella   nome
                 v - visualizza nomi
                 e - elimina   lista
                 u - USCITA

digitare scelta:
v

---contenuto Lista---
ATTENZIONE!!! Lista vuota, pTesta=0

digitare scelta (i/c/v/e/u):
i luca
ind.nuovo Nodo:0xc76cd8
i anna
ind.nuovo Nodo:0xc7a500
i zoe
ind.nuovo Nodo:0xc7a528
i ugo
ind.nuovo Nodo:0xc7a550
i aldo
ind.nuovo Nodo:0xc7a578
v
```

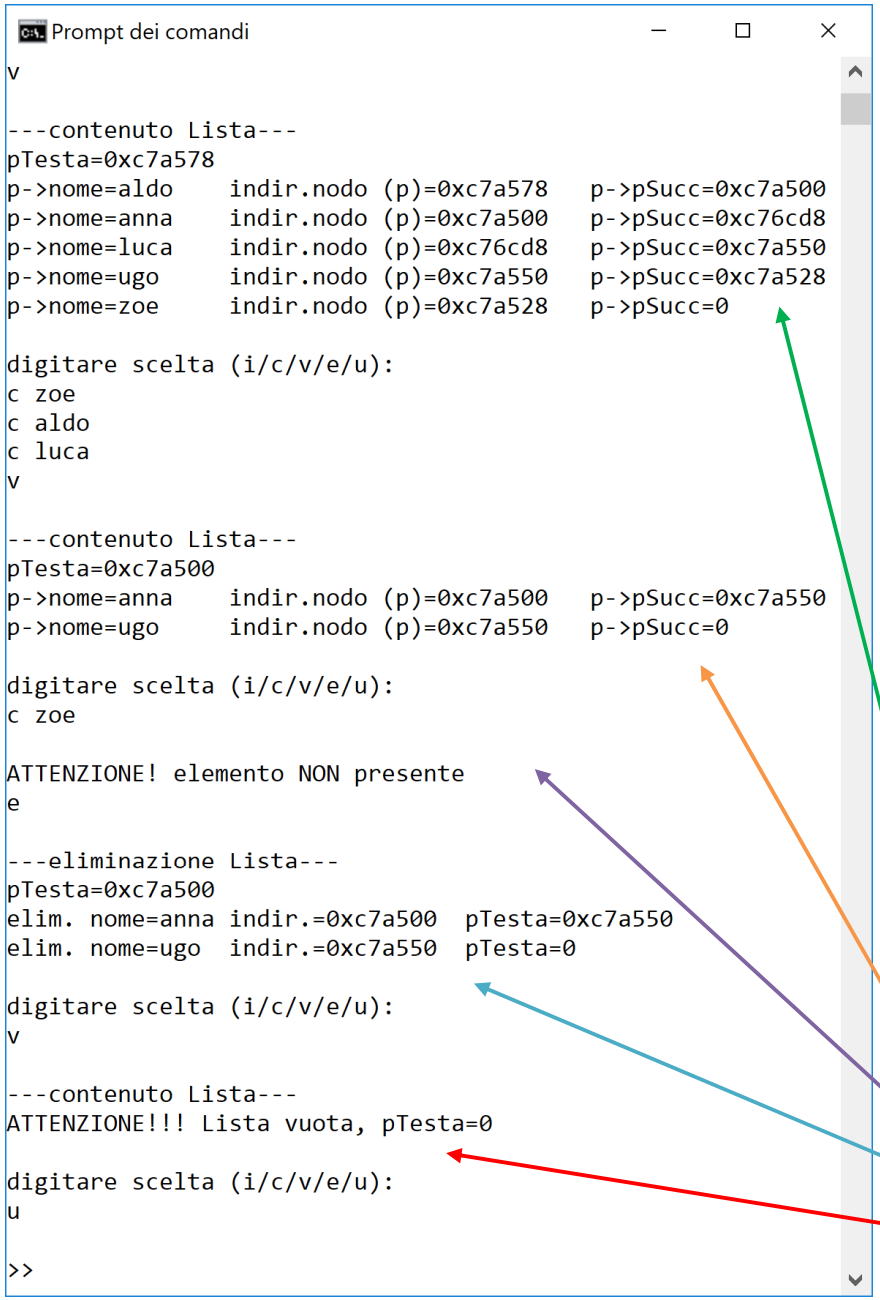
1) l'inserimento del nome ***luca*** determina l'allocazione di uno spazio di M.C. per il Nuovo Nodo all'indirizzo 0x**C76CD8**, indirizzo che viene salvato nel puntatore pTesta:
   `pTesta` vale 0x**C76CD8**
   `pTesta->nome` vale ***luca*** `pTesta->pSucc` vale ***NULL***
2) l'inserimento del nome ***anna*** determina l'allocazione di uno spazio di M.C. per il Nuovo Nodo all'indirizzo 0x**C7A500**, indirizzo che viene salvato nel puntatore pTesta (perché il nome è < pTesta->nome):
   `pTesta` vale 0x**C7A500**
   `pTesta->nome` vale ***anna*** `pTesta->pSucc` vale 0x**C76CD8**
   (il nodo di ***luca*** continua ad avere pSucc valorizzato a ***NULL***)

```
void fIns(){
   pNew        = new Nodo;      //--- allocazione Nuovo Nodo in M.C.
   pNew->nome  = nomeInp;       //--- valorizza campi Nodo
   pNew->pSucc = NULL;          //--- inizialmente a NULL
   cout<<"ind.nuovo Nodo:"<<pNew<<endl;

   if (pTesta == NULL)          //--- Lista Vuota  (caso 1. luca)
       pTesta = pNew;           //--- pTesta punta al Nuovo Nodo
   else
   if (pNew->nome  <  pTesta->nome){
//--- se Nuovo nome precede nome a cui punta pTesta (caso 2.anna e 5.aldo)
       pTemp          = pTesta; //--- salva indiriz. cui punta pTesta
       pTesta         = pNew;   //--- pTesta ora punta al Nuovo Nodo
       pTesta->pSucc = pTemp;   //--- aggiorna indir. cui punta Nuovo
   }       //--- il Nodo successivo di New ora è quello cui puntava pTesta
   else {  //--ricerca primo Nodo con nome MAGGIORE del Nuovo nome
       pTemp          = pTesta;
       while (pTemp     != NULL                    //--- caso 3. zoe
          && pNew->nome  >=  pTemp->nome) //caso 4. ugo
       {      pPrec       = pTemp;
              pTemp       = pTemp->pSucc; //--- punta a successivo
       }                  //--- seguendo la catena dei puntatori pSucc
       pNew->pSucc  = pTemp;    //--- Nuovo punterà al MAGGIORE
       pPrec->pSucc  = pNew;
   }                  //--- il precedente del MAGGIORE punterà a Nuovo
}
void fLisVuota(){
cout<<"ATTENZIONE!!! Lista vuota, pTesta=" << pTesta<<endl;}

void  fElim() {       //--- da pTesta seguendo la catena dei puntatori
  cout<< "\n---eliminazione Lista---\n";
  if (pTesta == NULL)     fLisVuota();     //--- Lista Vuota
  else {
    cout<<"\npTesta=" << pTesta << endl;
    while (pTesta != NULL){
       cout<<"elim. nome=" << pTesta->nome
           << "\tindir.="  << pTesta;
       pTemp  = pTesta;
       pTesta = pTesta->pSucc;       //--- pTesta deve puntare al
                                     //--- Nodo cui puntava il nodo di pTesta
       delete   pTemp;       //--- libera il Nodo cui puntava pTesta
       cout<< "  pTesta=" << pTesta << '\n';
     }
  } cout<< "\ndigitare scelta (i/c/v/e/u):\n";
}
```

```
void fCan(){
  bool trovato = false;
   if (pTesta == NULL)     fLisVuota();      //--- Lista Vuota
   else {
     if (pTesta->nome  ==  nomeInp){
                  //--- il nome cercato è nel Nodo puntato da pTesta
           trovato = true;
           pTemp   = pTesta->pSucc;
           delete   pTesta;
           pTesta  = pTemp;      //--- pTesta punterà al successivo
     } else {          //--- ricerca primo Nodo con nome richiesto
        pTemp         = pTesta->pSucc;
        pPrec         = pTesta;
        while (pTemp != NULL   &&   ! trovato) {
            if (pTemp->nome  ==  nomeInp){
                          //--- trovato il Nodo, il Nodo precedente
                trovato = true;         //--- deve puntare al Nodo
                pPrec->pSucc = pTemp->pSucc;   //--- cui puntava
                delete  pTemp;                 //--- il Nodo cancellato
            } else {                           //--se nome diverso
                pPrec = pTemp;      //---si passa al Nodo successivo
                pTemp = pTemp->pSucc; } //---seguendo la catena
        }                                  //--- dei puntatori pSucc
      }
      if (! trovato)
          cout<<"\nATTENZIONE! elemento NON presente\n";
   }
}
void  fVis() {
      cout<< "\n---contenuto Lista---\n";
      Nodo* p;
      p = pTesta;
      if (pTesta == NULL)     fLisVuota();      //--- Lista Vuota
      else {
        cout<<"pTesta=" << pTesta << endl;
        while (p != NULL){
          cout<<"p->nome="           << p->nome
              <<"\tindir.nodo (p)=" << p
              <<"   p->pSucc="       << p->pSucc << '\n';
          p = p->pSucc;          //--- si scorre la LISTA seguendo
        }                       //--- la catena dei puntatori pSucc
      }  cout<< "\ndigitare scelta (i/c/v/e/u):\n";
}
```



3) l'inserimento del nome ***zoe*** determina l'allocazione di uno spazio di M.C. per il Nuovo Nodo all'indirizzo 0x**C7A528**, indirizzo che viene salvato nel puntatore (pSucc) del Nodo che contiene il nome ***luca***. A partire da pTesta, infatti, si ricerca il primo Nodo che contenga un nome MAGGIORE di ***zoe***; **nessun Nodo soddisfa la condizione** ed il ciclo *while* termina per **pTemp = NULL** sul nodo di ***luca*** (il cui pSucc vale NULL) quindi:
   `pTesta` resta a 0x**C7A500**
   `pPrec->pSucc` (di ***luca***) = **pNew** (punterà al Nodo di ***zoe***)
   pNew`->pSucc` (di ***zoe***) = `pTemp` (cioè **=*NULL***, visto che in pTemp era salvato pSucc di ***luca***)
4) l'inserimento del nome ***ugo*** determina l'allocazione di uno spazio di M.C. per il Nuovo Nodo all'indirizzo 0x**C7A550**, indirizzo che viene salvato nel puntatore (pSucc) del Nodo che contiene il nome ***luca***. A partire da pTesta, infatti, si ricerca il primo Nodo che contenga un nome MAGGIORE di ***ugo***; il Nodo di ***zoe*** soddisfa la condizione ed il ciclo *while* termina perchè **pNew->nome** (***ugo***) **< pTemp->nome** (***zoe***). I valori di pPrec e pTemp salvati nel ciclo precedente sono:
   `pPrec = pTemp;` (0x**C76CD8** indirizzo ***luca***)
   `pTemp = pTemp->pSucc` (0x**C7A528** indirizzo ***zoe***)
   Quindi:
   `pTesta` vale sempre 0x**C7A500**
   `pPrec->pSucc(di` ***luca***`)=` **pNew** (punterà al Nodo di ***ugo***)
   pNew`->pSucc(di` ***ugo***`)= pTemp` (punterà al Nodo di ***zoe***)
5) per il nome ***aldo*** l'elaborazione è analoga a quella per ***anna*** (punto 2) Il nuovo Nodo ha indirizzo 0x**C7A578** e viene salvato nel puntatore pTesta.

Dopo i 5 inserimenti in LISTA, si richiede la **visualizzazione** che presenta la videata riprodotta a sinistra. Si procede poi con:

a) la **cancellazione** di *zoe, aldo* e *luca* e alla successiva **visualizzazione**;
b) la **cancellazione** del nome *zoe* (che è stato già cancellato) che determina una segnalazione di errore;
c) l'**eliminazione** di tutta la Lista (*anna* e *ugo*);
d) una successiva richiesta di **visualizzazione** che determina una segnalazione di errore.