



```

//===== Programma principale
int main()
{
    int scelta;
    elemento *p;
    string x;

    // inizializza la lista (puntatore alla testa della lista vuota)
    InizializzaLista();

do
{
    // presenta il menu di scelta
    menu();
    cout<<" Inserisci la tua scelta: ";
    cin>>scelta;

    switch(scelta)
    {
        case 1: InserisciInTesta();    // crea un nuovo elemento e lo inserisce in testa alla lista
            break;
        case 2: EliminaInTesta();      // cancella il primo elemento e libera la memoria
            break;
        case 3: InserisciInCoda();     // crea un nuovo elemento e lo inserisce in coda alla lista
            break;
        case 4: EliminaInCoda();       // cancella l'ultimo elemento e libera la memoria
            break;
        case 5: VisualizzaLista();     // visualizza tutti gli elementi della lista
            break;
        case 6: SvuotaLista();         // cancella tutti gli elementi della lista deallocando la memoria
            break;
        case 7: cout <<"\nInserisci il nome del prodotto da cercare : "; // Ricerca per Nome prodotto
            cin >>x;
            p = CercaElemento (x);    // p= puntatore all'elemento se è stato trovato (NULL=NON TROVATO)
            if (p!=NULL) VisualizzaDatiElemento(p);
            else cout<<x<<" NON TROVATO\n\n";
            break;
        case 8: cout <<"\nInserisci Nome prodotto per modifica dati : "; // Modifica dati di un prodotto
            cin >>x;
            p = CercaElemento (x);
            if (p!=NULL)
                AcquisisciDatiElemento(p);
            else
                cout<<x<<" NON TROVATO\n\n";
            break;
        case 0: SvuotaLista();
            break;
        default: cout <<"Scelta errata \n\n ";
    }
}while(scelta!=0);
}
// =====

```

## // FUNZIONI

### // --- presenta il menu

```
void menu()
{
    cout<<"          MENU' di GESTIONE LISTA DI PRODOTTI";
    cout<<'\n';
    cout<<"          1 - Inserisci elemento in testa\n";
    cout<<"          2 - Cancella elemento in testa\n";
    cout<<"          3 - Inserisci elemento in coda\n";
    cout<<"          4 - Cancella elemento in coda\n";
    cout<<"          5 - Visualizza elementi della lista\n";
    cout<<"          6 - Svuota tutta la lista\n";
    cout<<"          7 - Ricerca elemento per nome prodotto\n";
    cout<<"          8 - Modifica dati di un prodotto\n";
    cout<<"          0 - FINE\n";
    cout<<'\n';
}
```

### // --- Imposta il puntatore di testa a NULL

```
void InizializzaLista()
{ testa=NULL; // lista vuota
}
```

### // --- Controlla se la lista è vuota

```
bool ListaVuota()
{ if (testa==NULL)
    return true;
  else
    return false;
}
```

### // --- Svuota lista deallocando la memoria

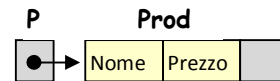
```
void SvuotaLista()
{elemento *temp; // puntatore d'appoggio
 // dealloca tutti gli elementi
 while (testa != NULL)
   {temp = testa;
    testa = testa->succ;
    delete temp; // dealloca l'elemento
   }
 cout <<"\n---- LISTA CANCELLATA\n\n";
}
```

### // --- Crea un nuovo elemento e restituisce il puntatore

```
elemento *CreaElemento()
{elemento *p;
 p=new (elemento); // crea il nuovo elemento
 return p; // restituisce il puntatore al nuovo elemento creato
}
```

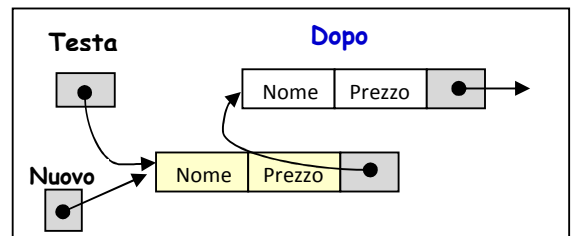
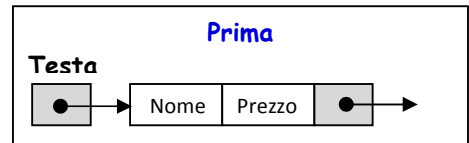


```
// --- Acquisisci da tastiera i dati dell'elemento della lista puntato da p
void AcquisisciDatiElemento(elemento *p)
{ cout <<"Inserisci Nome del prodotto : ";
  cin >>p->Prod.Nome;
  cout <<"Inserisci il prezzo : ";
  cin >>p->Prod.Prezzo;
  return;
}
```



```
// --- Inserimento di un nuovo elemento in testa
void InserisciInTesta( )
{
  elemento *nuovo=CreaElemento(); // crea un nuovo elemento
  AcquisisciDatiElemento(nuovo); // acquisisce i valori della parte Informazione

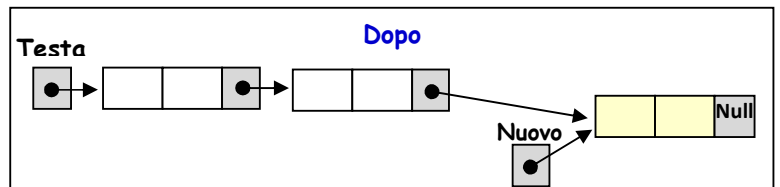
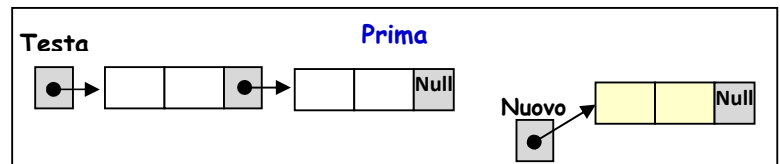
  nuovo->succ = testa;
  testa = nuovo;
  return;
}
```



```
// --- Inserimento di un nuovo elemento in coda
void InserisciInCoda()
{
  elemento *p, *nuovo;
  nuovo=CreaElemento(); // crea un nuovo elemento
  AcquisisciDatiElemento(nuovo); // acquisisce i valori del campo Informazione

  nuovo->succ = NULL; // assegna la marca di fine lista

  if (ListaVuota()) // Se la lista è vuota diventa il primo elemento e l'ultimo della lista
    testa = nuovo;
  else
  {p = testa; // altrimenti scorre la lista
   while (p->succ!=NULL)
     p = p->succ;
   p->succ=nuovo;
  }
}
```



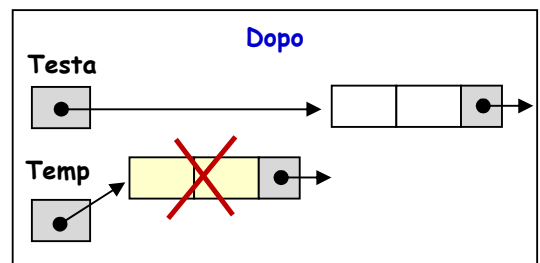
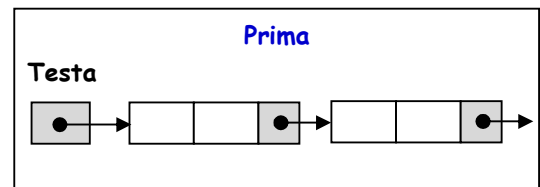
```
// --- Visualizza i dati dell'elemento della lista puntato da p
void VisualizzaDatiElemento(elemento *p)
{cout <<left<<setw(15)<<p->Prod.Nome<<right<<setw(10)<<p->Prod.Prezzo<<endl;
  return;
}
```

// --- Visualizza le informazioni contenute nella lista

```
void VisualizzaLista ()
{if (ListaVuota())
    cout<<"--- LISTA VUOTA\n";
else
    {elemento *p=testa;
    cout<<"---- Elenco prodotti ----\n";
    while (p!=NULL)
        {
            VisualizzaDatiElemento( p );
            p=p->succ;
        }
    cout<<"---- Fine Elenco -----\n\n";
}
}
```

// --- Elimina elemento in testa

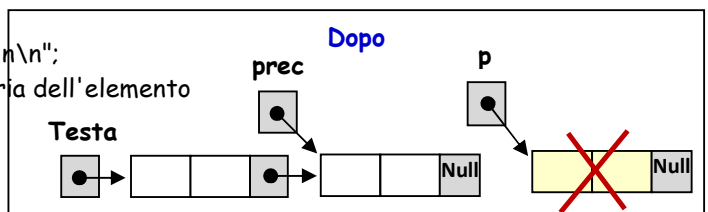
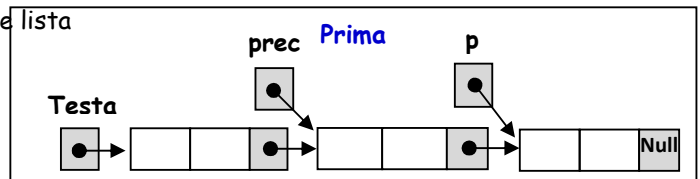
```
void EliminaInTesta()
{if (ListaVuota())
    cout<<"--- LISTA VUOTA\n";
else
    {elemento *temp=testa; // libera il primo elemento
    testa=temp->succ;
    cout <<"Prodotto "<<temp->Prod.Nome<<" cancellato\n\n";
    delete temp;
}
}
```



// --- Elimina elemento in coda

```
void EliminaInCoda()
{if (ListaVuota()) // lista vuota
    cout<<"--- LISTA VUOTA\n";
else
    {elemento *prec, *p=testa; // prec puntatore al precedente rispetto a p
    // p puntatore all'elem. attuale
    while (p->succ!=NULL) // Scorre fino a fine lista
        {prec = p;
        p=p->succ;
        }
    if (p == testa) // lista di un solo elemento
        testa = NULL; // allora cambia la testa della lista
    else
        prec->succ=NULL; // il precedente diventa l'ultimo della lista

    cout <<"Elemento "<<p->Prod.Nome<<" cancellato\n\n";
    delete p; // libera l'area di memoria dell'elemento
}
return;
}
```



```
// --- Ricerca per nome del prodotto
elemento *CercaElemento (string x) // Restituisce il puntatore all'elemento (NULL = non trovato)
{ if (ListaVuota()) return NULL; // Lista vuota
  else
  { elemento *p = testa;
    bool trovato=false;
    while (p !=NULL && !trovato)
    { if (p->Prod.Nome == x)
      trovato=true;
      else
      p=p->succ;
    }
    return p;
  }
}
```