

```

1: #include <iostream>
2: #include<malloc.h>
3:
4: using namespace std;
5:
6:
7: /* Definizione tipo nodo */
8: typedef struct elemento_lista{
9:     int valore;
10:    struct elemento_lista *succ;
11: } nodo;
12:
13:
14: typedef nodo *lista; /* Definizione tipo lista di interi */
15:
16: lista inserisciInTesta(lista l, int el){
17:     lista L;
18:     L = (lista) malloc (sizeof(nodo));
19:     L->valore = el;
20:     L->succ = l;
21:     return L;
22: }
23:
24: lista inserisciInCoda(lista l, int el){
25:     lista L, temp;
26:     if(l==NULL)
27:         {temp=(lista) malloc(sizeof(nodo));
28:           temp->valore=el;
29:           temp->succ= NULL;
30:           return temp;
31:         }
32:     else
33:         {for(temp=l; temp->succ!=NULL; temp=temp->succ);
34:           temp->succ = (lista) malloc (sizeof(nodo));
35:           temp->succ->valore = el;
36:           temp->succ->succ = NULL;
37:           return l;
38:         }
39: }
40:
41: int main(){
42:
43:     lista M,L,temp; /* creazione della lista */
44:
45:     L = (lista) malloc (sizeof(nodo)); /* inserimento primo elemento */
46:     L->valore = 1;
47:     L->succ = NULL;
48:
49:     M=inserisciInTesta(L, 12); /* inserimento secondo elemento */
50:     L=inserisciInTesta(M, 88);
51:     M=inserisciInTesta(L, 33);
52:     L=inserisciInTesta(M, 44);
53:     M=inserisciInCoda(L, 55);
54:     //cout<<"["<<L->valore<<"] ";
55:
56:     for(temp=L; temp->succ!=NULL; temp=temp->succ) cout<<temp->valore<<"->"; //Si procede nella
57:     cout<<temp->valore;
58:

```

```
59: return 0;  
60:  
61: }
```