

```
1  #include<iostream>
2  using namespace std;
3
4  // Định nghĩa NODE
5  struct NODE {
6      int data;
7      NODE* next;
8  };
9
10 // Định nghĩa LIST
11 struct LIST {
12     NODE* head;
13 };
14
15 NODE* tao_node(int);
16 void them_dau_danh_sach(LIST&, int);
17 void in_danh_sach(LIST);
18 int tinh_tong(LIST);
```

```

20 void main()
21 {
22     // Khai báo LIST
23     LIST l;
24     l.head = NULL;
25
26     // Thêm phần tử vào danh sách
27     int n;
28     cout << "Nhap so luong phan tu: ";
29     cin >> n;
30
31     int i = 0;
32     while (i < n)
33     {
34         int tam;
35         cout << "Nhap phan tu thu " << i + 1 << ": ";
36         cin >> tam;
37
38         // Thêm phần tử vào đầu danh sách
39         them_dau_danh_sach(l, tam);
40
41         i++;
42     }
43
44     cout << "Danh sach vua nhap: ";
45     in_danh_sach(l);
46     cout << endl;
47
48     cout << "Tong: " << tinh_tong(l) << endl;
49
50     system("pause");
51 }

```

```

53  NODE* tao_node(int x)
54  {
55      NODE* p;
56      p = new NODE;
57
58      if (p == NULL)
59      {
60          cout << "Khong the tao node";
61          return NULL;
62      }
63
64      p->data = x;
65      p->next = NULL;
66
67      return p;
68  }

```

```

70 void them_dau_danh_sach(LIST& l, int x)
71 {
72     // Tạo node
73     NODE* p = tao_node(x);
74     if (p == NULL)
75     {
76         cout << "Khong the tao NODE";
77         return;
78     }
79
80     // Thêm node p vào đầu danh sách l
81     // Danh sách rỗng
82     if (l.head == NULL)
83     {
84         l.head = p;
85     }
86     else // Danh sách không rỗng
87     {
88         p->next = l.head;
89         l.head = p;
90     }
91 }

--
93 void in_danh_sach(LIST l)
94 {
95     NODE* p = l.head;
96     while (p != NULL)
97     {
98         cout << p->data << " ";
99         p = p->next;
100     }
101 }

```

```
103 int tinh_tong(LIST l)
104 {
105     NODE* p = l.head;
106     int tong = 0;
107     while (p != NULL)
108     {
109         tong += p->data;
110         p = p->next;
111     }
112     return tong;
113 }
```