

```
1  /// Longest Increasing Subsequence(LIS)  $O(n^2)$ :
2  #include<bits/stdc++.h>
3  using namespace std;
4  vector<int>ed[1005],vv;
5  int n,dp[1005],a[1005],w[1005];
6  int lis(int u){
7      if(ed[u].size()==0)return dp[u]=1;
8      if(dp[u]!=-1)return dp[u];
9
10     int ret = 0;
11     for(int i=0; i<ed[u].size(); i++){
12         int v = ed[u][i];
13         ret = max(ret,1+lis(v));
14     }
15     return dp[u] = ret;
16 }
17 void path(int u,int x){
18     if(x==0)return;
19
20     for(int i=0; i<ed[u].size(); i++){
21         int v = ed[u][i]; int ret = 1+lis(v);
22         if(ret==x){
23             vv.push_back(a[v]);
24             path(v,x-1);
25             break;
26         }
27     }
28 }
29 int main(){
30     while(scanf("%d",&n)==1){
31         for(int i=1; i<=n; i++){
32             scanf("%d",&a[i]);
33         }
34
35         for(int i=1; i<=n; i++){
36             for(int j=i+1; j<=n; j++){
37                 if(a[j]>a[i]) {
38                     ed[i].push_back(j);
39                 }
40             }
41         }
42
43         for(int i=1; i<=n; i++) ed[0].push_back(i);
44
45         memset(dp,-1,sizeof(dp));
46         int ans = lis(0);
47
48         path(0,ans);
49
50         printf("%d:",ans-1);
51         for(int i=0; i<vv.size(); i++) printf(" %d",vv[i]);
52         printf("\n");
53
54         vv.clear(); for(int i=0; i<=n; i++)ed[i].clear();
55     }
56     return 0;
57 }
```