```
#include<bits/stdc++.h>
                          #define DIST(x1,x2, y1, y2) sqrt(((x1-x2)*(x1-x2))+((y1-y2)*(y1-
                          y2)))
                          #define DIST3D(x1,x2, y1, y2, z1, z2) (((x1-x2)*(x1-x2))+((y1-
                          y2)*(y1-y2)) + ((z1-z2)*(z1-z2)))
                          #define CLR(a) a.clear()
                          #define VCLR(a, n) for(int i=0; i<=n+3; i++) a[i].clear()
                          #define SIZE(a) a.size()
                          #define ERASE(a, b) memset(a, b, sizeof a)
                          #define pb push_back
                          #define LL long long
                          #define ULL unsigned long long
                          #define DBG cout<<"I Am Here"<<endl
                          #define DBGA(a) cout<<a<<endl
                          #define DBGI(b,a) cout<<b<<' '<<a<<endl
                          #define DBGL(i,s,e,b) or(int i=s; i<=e; i++) cout<<b<<endl
                          #define INF 1e9
                          #define INV 1e-6
                          #define SF(a) scanf("%I64d", &a)
                          #define PF(a) printf("%I64d\n", a)
                          #define sf(a) scanf("%d", &a)
                          #define pf(a) printf("%d\n", a)
                          #define pii pair<int,int>
                          #define PIS pair<double,string>
                          #define PII pair<LL,LL>
                          #define MAX 600005
                          #define CASE(i) printf("Case %d:", i);
                          #define PI acos(-1)
                          #define piis pair<int, string>
                          #define fast1 ios_base::sync_with_stdio(false);
                          #define fast2 cin.tie(0)
                          #define CHECK_BIT(var,pos) ((var & (1 << pos)) == (1 << pos))
                          #define LOOP(i, b, n) for(int i=b; i<=n; i++)
                          #define nl puts("")
                          #define popcount __builtin_popcount
                          #define valid(i,j,m,n) (i>=0 && i<n && j>=0 && j<m)
                          #define all(v) v.begin(), v.end()


                          using namespace std;
```

```
/** -----------------------------------------------**/
/** Header And Definitions Ends Here.             **/
/** ----------------------------------------------**/


int dx4[] = {0, 0, 1, -1}; int dy4[] = {1, -1, 0, 0};
int dx8[] = {0, 0, 1, -1, 1, 1, -1, -1}; int dy8[] = {1, -1, 0,
0, 1, -1, 1, -1};
int dxH[] = {2, 2, -2, -2, 1, 1, -1, -1}; int dyH[] = {1, -1, 1,
-1, 2, -2, 2, -2};


const double GRS = (1 + sqrt(5))/2;


template<typename T> T power(T X, T P)
{
    T ans = (T)1;
    for(T i=1; i<=P; i++){
        ans = ans * X;
    }
    return ans;
}


template<typename T> T ABS(T A, T B)
{
    T ret = A - B;
    if(ret<0) return -ret;
    return ret;
}


const LL MOD = 1000000007;
const LL BIGMAX = power(2,63) - 1;


template<typename T> T ADD(T X, T Y, T M)
{
    if(X+Y<0)
        return (X - M) + Y;
    else if(X+Y>=M)
        return X+Y-M;
    else
        return X+Y;
}


template<typename T> T prod(T a, T b, T c) // CUSTOM PRODUCT
FUNCTION FOR BIG NUMBER MULTIPLICATION
{
    T x = 0, y=a%c;
    while(b > 0){
        if(b%2 == 1){
            x = ADD(x, y, c);
        }
        y = (y*2)%c;
        b /= 2;
    }
    return x%c;
}


template<typename T> T bigmod(T a, T b, T c){
    T x = (T)1, y = a%c;
    while(b > 0) {
        if(b%(T)2 == (T)1) {
            x = (x * y)%c;
        }
        y = (y * y)%c;
        b /= (T)2;
    }
    return x;
}




template <typename T> T MODINVERSE(T a){
    return bigmod(a, MOD-2);
}


template<typename T> T GCD(T x, T y) {
  while ( y != 0 ) {
    T z = x % y;
    x = y;
    y = z;
  }
  return x;
}


bool isvowel(char ch)
{
    ch = toupper(ch);
    if(ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U' )
return true;
    return false;
}


template<typename T>T isleap (T year)
{
    if (year%(T)400 == (T)0 || (year%(T)100 != (T)0 && year%(T)4
== (T)0)) return true;
    return false;
}


/**--------------------------------------------------------------
--------------**/
/** Template Ends Here. Main Function And User Defined Functions
Starts Here. **/
/**--------------------------------------------------------------
------------**/


int arr[100005];
int Temp[100005], Result[100005];
int Val[100005];
int lislen;
vector<int>path;


// Temp Array will store Minimum of Last Value of an Increasing
Subsequence of Particular Length
```

```
int main()
{


    int n;
    scanf("%d", &n);
    for(int i=0; i<n; i++){
        scanf("%d", arr+i);
    }


    lislen = 0;
    ERASE(Result, -1);


    int i, j, k, l;
    int sz = 0;
    for(i=0; i<n; i++){
        if(i==0){
            Temp[sz] = i;
            Val[sz] = arr[Temp[sz]];
            sz++;
        }
        else{
            int idx = lower_bound(Val, Val+sz, arr[i]) - Val;
            if(idx==sz){
                Temp[sz] = i;
                Val[sz] = arr[Temp[sz]];
                Result[i] = Temp[sz-1];
                sz++;
                lislen++;
            }
            else{
                Temp[idx] = i;
                Val[idx] = arr[Temp[idx]];
                if(idx!=0){
                    Result[i] = Temp[idx-1];
                }
            }
        }
    }
    printf("LIS Length is: %d\n", lislen+1);
    for(int i=0; i<n; i++){
        printf("%d ", Result[i]);
    }
    nl;


    int idx = Temp[sz-1];
    path.push_back(arr[idx]);
    while(Result[idx]!=-1){
        path.push_back(arr[Result[idx]]);
        idx = Result[idx];
    }
    for(i=path.size()-1; i>=0; i--){
        printf("%d", path[i]);
        if(i>0) printf(" -> ");
    }
}
```