```cpp
/****88 segment tree min max ***/
#include <bits/stdc++.h>
#define mx 100023
using namespace std;
int arr[mx],arr1[mx];
int tree[mx * 3][2];
int n;
struct hi
{
    int m;
    int n;
}hol[100002]
;
bool comp(hi a,hi b)
{
    if(a.m!=b.m)
        return a.m<b.m;
    return a.n<b.n;
}
void init(int node, int b, int e)
{
    if (b == e) {
        tree[node][0] =1000000;
        tree[node][1]=-1;


        return;
    }
    int Left = node * 2;
    int Right = node * 2 + 1;
    int mid = (b + e) / 2;
    init(Left, b, mid);
    init(Right, mid + 1, e);
    tree[node][0] =
min(tree[Left][0],tree[Right][0]);
    tree[node][1] =
max(tree[Left][1],tree[Right][1]);
}
int query(int node, int b, int e, int l,int i, int
j)
{   int res;
    if ((i > e || j < b) && l==2)
        return 0;
    if ((i > e || j < b) && l==1)
        return 1000000;     //????? ??? ??????
    if (b >= i && e <= j && l==1)
        return tree[node][0]; //????????? ????????
    if (b >= i && e <= j && l==2)
        return tree[node][1]; //????????? ????????
    int Left = node * 2; //???? ????? ???
    int Right = node * 2 + 1;
    int mid = (b + e) / 2;
    int p1 = query(Left, b, mid, l,i, j);
    int p2 = query(Right, mid + 1, e, l,i, j);
    if(l==1)
    {
        res=min(p1,p2);
    }
    else
        res=max(p1,p2);
    return res;

     //??? ??? ??? ????? ??????
}
void update(int node, int b, int e, int i, int
```

```
newvalue)
 64  {
 65      if (i > e || i < b)
 66          return; //????? ??? ??????
 67      if (b >= i && e <= i) {
 68              tree[node][0]=newvalue;
 69              tree[node][1]=newvalue;
 70          return;
 71      }
 72      int Left = node * 2; //???? ????? ???
 73      int Right = node * 2 + 1;
 74      int mid = (b + e) / 2;
 75      update(Left, b, mid, i,newvalue);
 76      update(Right, mid + 1, e, i,newvalue);
 77      tree[node][0] =
min(tree[Left][0],tree[Right][0]);
 78      tree[node][1]=
max(tree[Left][1],tree[Right][1]);
 79  }
 80  int y[100001][3];
 81  map<int,int>m;
 82  int main()
 83
 84  {  int a,b,c,d,e,i,j,p,q;
 85     int arr[100001],z[100001];
 86
 87     cin>>a;
 88     for(i=1;i<=a;i++)
 89     {
 90         scanf("%d",&arr[i]);
 91         hol[i].m=arr[i];
 92         hol[i].n=i;
 93
 94
 95     }
 96
 97     sort(hol+1,hol+a+1,comp);
 98
 99     for(i=1;i<=a;i++)
100     {   m[i]=hol[i].m;
101         arr[hol[i].n]=i;
102
103     }
104
105
106     init(1,1,a);
107
108     update(1,1,a,arr[1],arr[1]);
109
110
111     for(i=2;i<=a;i++)
112     {
113         d=arr[i];
114         p=query(1,1,a,2,1,d);
115         q=query(1,1,a,1,d,mx*3);
116
117         if(p!=0 && y[p][2]==0)
118         {
119             y[p][2]=1;
120             z[d]=p;
121
122             update(1,1,a,d,d);
123             continue;
124
125         }
126         q=query(1,1,a,1,d+1,a);
```

```
127             y[q][1]=1;
128             z[q]=q;
129             update(1,1,a,d,d);
130 
131 
132         }
133         for(i=2;i<=a;i++)
134         {
135             printf("%d ",m[z[arr[i]]]);
136         }
137         return 0;
138     }
```