```
int pref_max (int p){
  int res=0;
  for(;p>0;p-=p&-p){
    res=max(res,fenw_max[p]);
  }return res;
}void update_max(int p,int x){
  for(;p<=m;p+=p&-p){
    fenw_max[p]=max(fenw_max[p],x);
  }
}void add(int x){
  int prv=0;
  for(int pos:where[x]){
    update_max(prv+1,pos);
    prv=pos;
  }
  index_lim=min(index_lim,prv);
}
int main(){
  ios::sync_with_stdio(0)
  cin.tie(0);
  cin>>n;
  for(int i=1;i<=n;i++){
    cin>>a[i];
    where[a[i]].pb(i)
  }
  index_lim=n;
  for(int x=0;x<=n-1;x++){
    add(x)
    int i=1,ans=0
    while(i<=n){
      int j;
      if(i>index_lim){
        j=m+1;
      }else{
        j=pref_max(i)
      }ans++
      in(j==i){
        ans=-1;
        break
      }i=j;
    }cout<<ans<<"\n"[x==n];
  }//cerr<<(double)clock()/
  CLOCKS_PER_SEC<<endl
  return 0;
}
```

2-бөлім

```
(//#pragma GCC optimize
("-Ofast")
#include<bits/stdc++.h>
#define pb push_back
#define all(x) (x).begin()
(x).end()
#define sz(x) (int)(x).size()
using namespace std;
typedef long long ll;
const int MAXN=(int)1e5+
const int K=300;
const int l=MAXN/K+2
int P[l][MAXN],a[l][l]
//S[MAXN],T[MAXN];
//a[MAXN],b[l]
int p[MAXN];
int n,m,q;
void sqrtUpd1(int p,ll x){
  S[p]+=x;
  T[p/K]+=x;
}
void sqrtUpd(int id,int l,
int x){
int ci=l/K,cn=n/K;
```