```cpp
/*** Z Algo ***********/
#include <bits/stdc++.h>
using namespace std;
int z[100002];
vector<int>v;
struct info {
    int prop, sum;
} tree[100002 * 3]; //sum ????? ???? ???????? ?? ???? ????? ???? ?????? prop ?
void update(int node, int b, int e, int i, int j, int x)
{
    if (i > e || j < b)
        return;
    if (b >= i && e <= j) //?????? ????? ??????? ??????? ?????
    {
        tree[node].sum += ((e - b + 1) * x); //???? ???? ??? e-b+1 ?? ??? e-b+1 ??? x ???? ??? ?? ??????
        tree[node].prop += x; //????? ?????????? ???? x ???? ???
        return;
    }
    int Left = node * 2;
    int Right = (node * 2) + 1;
    int mid = (b + e) / 2;
    update(Left, b, mid, i, j, x);
    update(Right, mid + 1, e, i, j, x);
    tree[node].sum = tree[Left].sum + tree[Right].sum + (e - b + 1) * tree[node].prop;
    //????? ???? ??? ???? ????????? ????? ???
    //??? ?? ??? ????? ??? ????? ???? ??? ???? ???????? ???? ???? ???
}
int query(int node, int b, int e, int i, int j, int carry = 0)
{
    if (i > e || j < b)
        return 0;

    if (b >= i and e <= j)
        return tree[node].sum + carry * (e - b + 1); //??? ?? ???? ???? ??? ??? ??????? ???? ???????? ?? ???? ???? ????? ????

    int Left = node << 1;
    int Right = (node << 1) + 1;
    int mid = (b + e) >> 1;

    int p1 = query(Left, b, mid, i, j, carry + tree[node].prop); //????????'?????? ???'???? ?????? carry ???????????
    int p2 = query(Right, mid + 1, e, i, j, carry + tree[node].prop);

    return p1 + p2;
}
int main()
{
int L = 0, R = 0,a,b,c,d,e,f,j,T,n;
char s[100002];
cin>>T;
for(j=1;j<=T;j++)
{   memset(z,0,sizeof(z));
    memset(tree,0,sizeof(tree));
    v.clear();
```

```
    L=0;
    R=0;
    scanf("%s",&s);
    n=strlen(s);


    for (int i = 1; i < n; i++) {
  if (i > R) {
    L = R = i;
    while (R < n && s[R-L] == s[R]) R++;
    z[i] = R-L; R--;
  } else {
    int k = i-L;
    if (z[k] < R-i+1) z[i] = z[k];
    else {
      L = i;
      while (R < n && s[R-L] == s[R]) R++;
      z[i] = R-L; R--;
    }
  }
}
z[0]=n;
f=0;
for(int i=0;i<n;i++)
    {
       if(z[i]!=0)
       {
          update(1,1,100001,1,z[i],1);
       }

    }
scanf("%d",&b);
for(int i=1;i<=b;i++)
{
    scanf("%d",&c);
    if(c!=0)
     {
      d=query(1,1,100001,c,c,0);
    v.push_back(d);
     }
     else
        v.push_back(n);

}
d=v.size();
for(int i=0;i<d;i++)
{
   printf("%d ",v[i]);
}
printf("\n");


}
return 0;
}
```