NAJPF - Pattern Find

[#kmp-algorithm](https://www.spoj.com/problems/tag/kmp-algorithm)

Your task is so simple given a string and a pattern. You find the pattern on the given string. If found print how many time found the pattern and their index. Otherwise print ‘Not Found’

**Input:**

The input   line consists of a number T (1 ≤ T ≤ 50) test cases.

For each case given two string number  A,B. the string and the pattern  1 ≤|A|, |B| ≤10^6

All character will be lower case Latin character.  And |  | is the length of string.

**Output:**

For each case print the number  (found pattern from the given string) next line there position And Otherwise print 'Not Found' without quota.  
There will a blank line between two cases.

**Sample:**

|  |  |
| --- | --- |
| Input | Output |
| 3 ababab ab aaaaa bbb aafafaasf aaf | 3 1 3 5  Not Found  1 1 |

Hints:

Here all index is 1 base.

[Submit solution!](https://www.spoj.com/submit/NAJPF/)

Solution:

1. #include<bits/stdc++.h>
2. #define pb push\_back
3. #define int long long int
4. #define INF 1e18
5. #define vec vector<int>
6. #define REP(i,a,b) for(i=a;i<b;i++)
7. using namespace std;
8. int mod=1e9+7;
9. main()
10. {
11. ios\_base::sync\_with\_stdio(false);
12. [cin](http://www.opengroup.org/onlinepubs/009695399/functions/cin.html).tie(NULL);
13. [cout](http://www.opengroup.org/onlinepubs/009695399/functions/cout.html).tie(NULL);
14. int t;
15. cin>>t;
16. while(t--)
17. {
18. string a,b;
19. cin>>a>>b;
20. int x=a.length(),y=b.length();
21. int i=1,j=0;
22. int h[y+1]={0};
23. while(i<y)
24. {
25. if(b[j]==b[i])
26. h[i]=1+h[j],i++,j++;
27. else
28. h[i]=0,i++,j=h[j];
29. }
30. vector<int> occ;
31. i=0,j=0;
32. while(i<x)
33. {
34. if(a[i]==b[j])
35. i++,j++;
36. else
37. {
38. if(!j)
39. i++;
40. if(j==h[j])
41. j=0;
42. j=h[j];
43. }
44. if(j>=y)
45. occ.pb(i-y),j=0;
46. }
47. if(occ.size()==0)
48. cout<<"Not Found**\n**";
49. else
50. {
51. cout<<occ.size()<<"**\n**";
52. for(int ele:occ)
53. cout<<ele+1<<" ";
54. cout<<"**\n**";
55. }
56. }
57. }