

Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
8192208	02:06:00	Suffix Array Re-construction	✔ Accepted	0.42 s	C++
	✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔✔				

Compiler output

```
./suffixarrayreconstruction.cpp: In function 'int main()':
./suffixarrayreconstruction.cpp:37:17: warning: ignoring return value of 'int scanf(const char*, ...)', declared with attribute warn_unused_result [-Wunused-result]
   37 |         int T; scanf("%d",&T);
      |         ~~~~~^~~~~~
./suffixarrayreconstruction.cpp:41:14: warning: ignoring return value of 'int scanf(const char*, ...)', declared with attribute warn_unused_result [-Wunused-result]
   41 |         scanf("%d %d",&L,&S);
      |         ~~~~~^~~~~~
./suffixarrayreconstruction.cpp:46:18: warning: ignoring return value of 'int scanf(const char*, ...)', declared with attribute warn_unused_result [-Wunused-result]
   46 |         scanf("%d %s",&p,in);
      |         ~~~~~^~~~~~
```

Submission contains 1 file: [download zip archive](#)

FILENAME	FILESIZE	SHA-1 SUM	
sl Help econstruction.cpp	1292 bytes	f7675f19df16bd51ca7dd13534cf40446f10f29e	download

Edit and resubmit this submission.

suffixarrayreconstruction.cpp

```
1 #include<bits/stdc++.h>
2 using namespace std;
3
4 int MX_LS = 10001;
5 int L,S;
6
7 char * in;
8 char * out;
9
10 bool check(int p){
11     int i = 0;
12     int n = strlen(in);
13     int cnt = 0;
14     for(i = 0; i < n; i++){
15         if(in[i]!='*')
16             cnt++;
17     }
18     if(cnt > L-p)
19         return false;
20     i=0;
21     while(i < n && in[i]!='*'){
22         if(out[i+p] != '_' && out[i+p] != in[i])
23             return false;
24         out[i+p]=in[i];
25         i++;
26     }
27     i = n-1;
28     while(i >=0 && in[i] != '*'){
29         if(out[L-(n-i)]!='_' && out[L-(n-i)]!=in[i])
30             return false;
31         out[L-(n-i)] = in[i];
32         i--;
33     }
34     return true;
35 }
36
37 int main(){
38     int T; scanf("%d",&T);
39     out = new char[MX_LS];
40     new char[MX_LS];
41     Help e(~--T){
42         scanf("%d %d",&L,&S);
43         memset(out, '_', L);
```

```
43     int p;  
44     bool fail = false;  
45     for(int i = 0; i < S; i++){  
46         scanf("%d %s",&p,in);  
47         fail |= !check(--p);  
48     }  
49     for(int i = 0; i < L && !fail; i++)  
50         if(out[i]=='_')  
51             fail=true;  
52     if(fail)  
53         puts("IMPOSSIBLE");  
54     else {  
55         out[L]='\0';  
56         puts(out);  
57     }  
58 }  
59 delete in;  
60 delete out;  
61 return 0;  
62 }
```