

7-2 堆栈操作合法性 (20 分)

假设以S和X分别表示入栈和出栈操作。如果根据一个仅由S和X构成的序列，对一个空堆栈进行操作，相应操作均可行（如没有出现删除时栈空）且最后状态也是栈空，则称该序列是合法的堆栈操作序列。请编写程序，输入S和X序列，判断该序列是否合法。

输入格式：

输入第一行给出两个正整数N和M，其中N是待测序列的个数，M（ $\leq 50$ ）是堆栈的最大容量。

随后N行，每行中给出一个仅由S和X构成的序列。序列保证不为空，且长度不超过100。

输出格式：

对每个序列，在一行中输出YES如果该序列是合法的堆栈操作序列，或NO如果不是。

输入样例：

```
4 10
SSSXXSXXSX
SSSXXSXXS
SSSSSSSSSXSSXXXXXXXXXX
SSSXXSXX
```

输出样例：

```
YES
NO
NO
NO
```

```
#include <stdio.h>
#include <malloc.h>
#include <string.h>
#define OK 1
#define STACK_INIT_SIZE 100
#define ERROR 0;
typedef char SElemType;
typedef int Status;
```

```
typedef struct
{
    SElemType *base;
    SElemType *top;
    int stacksize;
}SqStack;
```

```
int main(){
    int N;
    int M;
    int count=0;
    int cur=0;
```

```

if(scanf("%d %d",&N,&M)==2){}else{}
while(N--){
    SqStack s;
    s.base=(SElemType*)malloc(M*sizeof(SElemType*));
    s.top=s.base;
    s.stacksize=M;
    int flag=1;
    char ch[1000];

    scanf("%s",ch);
    int len=strlen(ch);
    for(int i=0;i<len;i++){
        if(ch[i]=='S'){
            *s.top++=ch[i];
            cur++;
            if(cur>s.stacksize){
                printf("NO\n");
                flag=0;
                break;
            }

        }else if(ch[i]=='X'){
            if(s.top==s.base){
                printf("NO\n");
                flag=0;
                break;
            }else{
                s.top--;
                cur--;
            }
        }

    }

    if(flag){
        if(s.top==s.base) printf("YES\n");
        else printf("NO\n");
    }

}

return 0;
}

```

