

# AcWing-4118：猫和狗

## 题目描述

你在动物收容所工作，负责喂养动物。

你一共准备了  
 $D$  份狗粮和  
 $C$  份猫粮。

一共有  $N$  只动物排队等候用餐，有的是狗，有的是猫。

当然，也有可能全都是狗或者全都是猫。

我们可以用一个长度为  
 $N$  的由大写字母  
 $C$  和  
 $D$  组成的字符串  
 $S$  来表示队列中猫狗的顺序。

如果队列中第  
 $i$  只动物是猫，则第  
 $i$  个字符为  
 $C$ 。

如果队列中第  
 $i$  只动物是狗，则第  
 $i$  个字符为  
 $D$ 。

动物们严格按照排队顺序依次进食。

每只狗吃一份狗粮，每只猫吃一份猫粮。

此外，你还有额外的猫粮。

每当一条狗吃完一份狗粮，你就会为猫多提供  $M$  份猫粮。

每只动物都只会在排在其前面的所有动物都进食完毕后，才肯进食。

这也就意味着，当轮到某只动物进食，但是却没有相应的食物时，它和排在它后面的所有动物都会因此无法进食。

难度：	简单
时/空限制：	1s / 64MB
总通过数：	1082
总尝试数：	3127
来源：	Google Kickstart2021 Round G Problem A
算法标签	▼

请问，在这种情况下，队列中的**所有狗**能否都得到喂食。

输入格式

第一行包含整数

$T$ ，表示共有

$T$  组测试数据。

每组数据第一行包含四个整数  $N, D, C, M$ 。

第二行包含一个长度为

$N$  的由大写字母

$C$  和

$D$  组成的字符串

$S$ 。

输出格式

每组数据输出一个结果，每个结果占一行。

结果表示为 `Case #x: y`，其中

$x$  为组别编号（从

1 开始），如果所有狗都能得到喂食，则

$y$  为 `YES`，否则

$y$  为 `NO`。

数据范围

$1 \leq T \leq 100$ ,

$1 \leq N \leq 10^4$ ,

$0 \leq D, C \leq 10^6$ ,

$0 \leq M \leq 10^6$ 。

输入样例1：

```
3
6 10 4 0
CCDCDD
4 1 2 0
CCCC
4 2 1 0
DCCD
```

输出样例1：

```
Case #1: YES
Case #2: YES
Case #3: NO
```

样例1解释

在 Case 1 中，一共有 10 份狗粮和 4 份猫粮。

- 1. 前两只动物是猫，喂食它们后，还剩下 2 份猫粮。
- 2. 接下来是一只狗，喂食它后，还剩下 9 份狗粮。
- 3. 然后是一只猫，喂食它后，还剩下 1 份猫粮。
- 4. 最后是两只狗，喂食它们后，还剩下 7 份狗粮。

所有狗都被喂食。

在 Case 2 中，没有狗，因此，所有狗（0 只）都被喂食了。

在 Case 3 中，第二只狗前面的猫得不到喂食，所有第二只狗也没法得到喂食。

输入样例2：

```
2
12 4 2 2
CDCCCDCCDCDC
8 2 1 3
DCCCCDC
```

输出样例2：

```
Case #1: YES
Case #2: NO
```

样例2解释

在 Case 1 中，每只狗喂食完毕后，都会额外得到两份猫粮。

- 1. 首先是一只猫，喂食它后，还剩下 1 份猫粮。
- 2. 接下来是一只狗，喂食它后，还剩下 3 份狗粮和 3 份猫粮。
- 3. 接下来是三只猫，喂食它们后，还剩下 3 份狗粮和 0 份猫粮。
- 4. 接下来是一只狗，喂食它后，还剩下 2 份狗粮和 2 份猫粮。
- 5. 接下来是两只猫，喂食它们后，还剩下 2 份狗粮和 0 份猫粮。
- 6. 接下来是一只狗，喂食它后，还剩下 1 份狗粮和 2 份猫粮。
- 7. 接下来是一只猫，喂食它后，还剩下 1 份狗粮和 1 份猫粮。
- 8. 接下来是最后一只狗，喂食它后，还剩下 0 份狗粮和 3 份猫粮。

所有狗都被喂食。

在 Case 2 中，第二只狗前面的猫得不到喂食，所有第二只狗也没法得到喂食。

算法思路

直接模拟即可。需要注意的是  $N \times M$  可能会超过 `INT_MAX`，所以需要使用 `long long int` 存储。

```
//
//  main.cpp
//  4118-狗和猫
//
//  Created by MacBook Pro on 2023/8/9.
//

#include <iostream>
```

```

using namespace std;

int main() {
    int T;
    scanf("%d",&T);
    for(int cases=1;cases<=T;cases++){
        long long N,D,C,M;
        scanf("%lld%lld%lld%lld",&N,&D,&C,&M);
        string s;
        cin>>s;
        //模拟猫狗进食
        bool flag=true;
        for(int i=0;i<N;i++){
            if(s[i]=='D'){
                if(D<=0){
                    flag=false;
                    break;
                }
                D--;
                C+=M;
            }
            else{
                if(C<=0){
                    //判断后面是否还有狗
                    bool is_dog=false;
                    for(int j=i+1;j<N;j++){
                        if(s[j]=='D'){
                            is_dog=true;
                            break;
                        }
                    }
                    if(is_dog){
                        flag=false;
                        break;
                    }
                    break;
                }
                C--;
            }
        }
        printf("Case #%d: ",cases);
        if(flag) printf("YES\n");
        else printf("NO\n");
    }
}

```