

TÌM BÒ

Cô bò Bessie đã trốn thoát và đang trốn ở một đồi núi với những đồng cỏ cao. Nông dân John (FJ), người đang muốn tìm kiếm Bessie, đã quyết định bò trên đồng cỏ bằng tay và đầu gối để tìm ra dấu vết của Bessie. Không may thay, ông ta có vấn đề với việc tìm kiếm Bessie từ vị trí thuận lợi này. Dãy cỏ ở trước mặt FJ trông như một chuỗi ngoặc đơn có độ dài N ($1 \leq N \leq 50,000$); ví dụ:

)(((())())

FJ biết rằng Bessie chân sau của Bessie giống như một cặp dấu mở ngoặc đơn ($($, và chân trước của cô ta giống như một cặp dấu đóng ngoặc đơn $)$. Vị trí của Bessie có thể được diễn tả bởi một cặp $x < y$, trong đó $($ được tìm ở vị trí x , và $)$ được tìm ở vị trí y . Hãy đếm có bao nhiêu vị trí mà Bessie có thể đang đứng.

Input

Dòng 1: Một chuỗi ngoặc đơn có độ dài là N ($1 \leq N \leq 50,000$).

Output

Dòng 1: Số vị trí mà Bessie có thể đứng – Có nghĩa là số cặp (x, y) khác nhau với $x < y$ sao cho $($ được tìm thấy ở x và $)$ được tìm thấy ở y .

Example

1

2

3

4

5

Input:

)(((())())

Output:

4

Có 4 vị trí có thể của Bessie được thể hiện ở dưới

1.)(((())())

^^ ^^

2.)(((())())

^^ ^^

3.)((()))()

^^ ^^

4.)((()))()

^^ ^^

#####

Hướng dẫn:

– Bài này sử dụng thuật toán đếm phân phối....

– Bài này chỉ cần lưu ý 1 chỗ, đó là nếu làm theo cách thông thường ta nghĩ ngay đến việc tìm cặp chân “(” rồi tìm xem ứng với mỗi cặp đó có bao nhiêu cặp “)” tương ứng có thể có...

– Nếu làm như trên sẽ dẫn tới TLE. Ở đây các bạn chỉ cần đếm số chân trước “)”, rồi lưu số chân trước đó vào 1 biến, tiếp theo sẽ xét từ trái qua phải, nếu gặp cặp “(” thì cộng số chân trước đã lưu vào tổng số vị trí có thể được, nếu gặp “)” thì trừ số chân trước đi 1, cứ làm tiếp tục như vậy đến khi cặp chân trước = 0.

Code tham khảo(pascal, C++):

[C++]

```
#include <string.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char a[50001];
```

```
    long count=0,tong=0;
```

```
    cin >> a;
```

```
    for(long i=0;i<strlen(a)-1;i++)
```

```
        if(a[i]==' '&& a[i+1]==' ')
```

```
            count++;
```

```
    for(long i=0;i<strlen(a);i++)
```

```
{
```

```
    if(a[i]=='(' && a[i+1]==')
```

```

        tong+=count;

        if(a[i]=='')&&a[i+1]=='')

            count--;

    }

    cout << tong;

}

```

[PASCAL]

```

const  fi="";

        nmax=50000;

type

        data=longint;

var

        f:text;

        s:ansistring;

        dem:data;

procedure xuli;

var    i,j:data;

        res:int64;

begin

        dem:=0;

        for i:=1 to length(s)-1 do

            if (s[i]='') and (s[i+1]='') then

                inc(dem);

        res:=0;

        for i:=1 to length(s)-1 do

            begin

                if (s[i]='(') and (s[i+1]='(') then

                    res:=res+dem;

```

```
        if (s[i]='') and (s[i+1]='') then
            dec(dem);
        end;
    writeln(res);
end;

begin
    assign(f,fi); reset(f);
    readln(f,s);
    close(f);
    xuli;
end.
```