

АВТОБУС

Решение на задачата с връщане назад изисква постоянна информация на коя спирка дали има качващи и слизащи.

Затова предварително, докато четем входните данни, в два булеви масива $f1$ и $f2$ ще отбелязваме тази информация.

Очевидно е, че автобусът спира само на спирка, за която има заявка за слизване и на спирката има заявка за качване.

На спирка s , на която има само чакащи пътници и няма желаещи да слязат, на нея няма да се спира ($f1[s]=true$, $f2[s]=false$). Тогава всички тези чакащи ще се приберат пеша.

На спирка t , за която има хора, които ще слизат, но няма чакащи, пак няма да спре автобусът ($f1[t]=false$, $f2[t]=true$). Тогава всички пътници, които искат да слязат на тази спирка t , ще слязат на следващата. Такава винаги има, защото по условие спира на последната спирка.

Докато прочитаме данните, запълваме и масивите $f1$ и $f2$.

Преди това можем да означим $f1[1]=f2[1]=f1[n]=f2[n]=true$, защото на спирка 1 винаги се качва пътник и се спира на нея. На спирка n се спира и $f2[n]=true$. Правим $f1[n]=true$, защото по алгоритъма, спирка на която $f1$ и $f2$ са $true$ се спира на нея.

Определяме bP – колко ще се приберат пеша и bS – колко ще слязат не на тяхната, а на следващата спирка.

```
for (i=1; i<=k-1; i++) {
    p1=c[i].p1; p2=c[i].p2;
    if (f1[p1]==true and f2[p1]==false) { // прибира се пеша
        bP++;
    }
    else
    if (f1[p1] and f2[p1]){                // Ако спира на p1
        if (!(f1[p2] and f2[p2]))        // Ако НЕ спира на p2
            bS++;                        // отива на следваща спирка
    }
}
```

Автор Павел Петров