

Ăn tối (Dinner - DINNER.*)

Trong một kì thi lập trình, ban tổ chức quyết định tài trợ bữa tối cho các thí sinh tham gia. Có tổng cộng N ($1 \leq N \leq 100$) thí sinh xếp thành một hàng dài để chờ ăn tối.

Trường ban tổ chức nhận ra rằng, các thí sinh rất ghét phải ngồi chung bàn ăn với một thí sinh sử dụng một ngôn ngữ lập trình khác. Kì thi chỉ có 2 loại ngôn ngữ lập trình: Gnode và Helpfile. Ngoài ra, để tiết kiệm số bàn cần phục vụ, nhà ăn yêu cầu một bàn phải có ít nhất K ($1 \leq K \leq 6$) thí sinh.

Trường ban tổ chức quyết định sẽ phân chia bàn ăn theo các bước sau:

Bước 1: Chọn một nhóm X thí sinh *liên tiếp* trong hàng và cho các thí sinh này vào 1 bàn, dĩ nhiên X không được bé hơn K và các thí sinh này phải sử dụng chung một ngôn ngữ lập trình.

Bước 2: Các thí sinh còn lại trong hàng sẽ dồn hàng lại, khóa lấp khoảng trống tạo ra bởi các thí sinh đã vào bàn ăn.

Bước 3: Nếu vẫn còn thí sinh chưa có bàn ăn, quay lại bước 1.

Yêu cầu: Hãy tính toán số bàn ăn tối thiểu cần có để tất cả các thí sinh có thể dùng bữa.

Dữ liệu: File văn bản DINNER.INP có:

- Dòng đầu chứa 2 số nguyên N và K như miêu tả đề bài phía trên.
- Dòng thứ 2 chứa 1 xâu gồm N kí tự miêu tả hàng thí sinh ban đầu. Kí tự thứ i trong xâu thể hiện ngôn ngữ lập trình của thí sinh thứ i trong hàng (G nếu là Gnode, H nếu là Helpfile).

Kết quả: Ghi ra file văn bản DINNER.OUT gồm duy nhất một số, là số bàn ăn ít nhất để tất cả thí sinh dùng bữa. Nếu không có cách sắp xếp thí sinh vào bàn ăn theo yêu cầu, in ra -1.

Ví dụ:

DINNER.INP	DINNER.OUT
7 2 GHHGHHG	3

Giải thích: Cho thí sinh thứ 2 và thứ 3 vào chung một bàn ăn, hàng thí sinh còn lại là GGHHG. Sau đó, cho thí sinh thứ 3 và thứ 4 vào chung một bàn ăn, hàng thí sinh còn lại là GGG. Cho cả 3 thí sinh còn lại vào một bàn ăn, như vậy số bàn ăn cần dùng là 3.