

Ăn tối (Dinner – bản full: DINNERF.*)

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 một bàn ($X \geq K$) 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 DINNERF.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 DINNERF.OUT như sau:

Dòng đầu gồm duy nhất một số T , 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 thì $T = -1$.

Khi $T > -1$, ghi ra $T+1$ dòng, từ dòng thứ 2 lần lượt là trạng thái ban đầu của hàng thí sinh và các trạng thái tiếp theo sau mỗi lượt xếp được 1 bàn ăn. Dấu “-” dùng để thể hiện thí sinh ở vị trí tương ứng đã rời khỏi hàng.

Ví dụ:

DINNERF.INP	DINNERF.OUT
7 2	4
GHHGHGGHG	GHHGHGGHG
	GHHGH--HG
	GHHG----G
	G--G----G

(Bản Full được đề xuất bởi Ngô Đức Minh – THPT Chuyên LHP, Tp.HCM)