

BLOG

Tên chương trình: BLOG.???

Harry và Rôn là người thường xuyên đọc blog của Steve. Theo yêu cầu của 2 bạn, Steve đã viết một chương trình chấm điểm. Harry rất thích nội dung của blog, Rôn thì ngược lại. Cách đây n ngày điểm của blog là x và hai bạn tranh luận với nhau xem sau n ngày điểm sẽ thay đổi như thế nào. Mỗi ngày nếu Harry bấm “+1” thì điểm đánh giá blog tăng thêm 1, nếu Rôn bấm “-1” thì điểm giảm đi 1. Chương trình chấm chưa hoàn thiện nên nếu một ngày nào đó cả 2 bạn cùng cho điểm thì điểm của blog sẽ là một số nguyên nào đó có giá trị tuyệt đối không vượt quá k .

Hôm nay Harry và Rôn quyết định chấm dứt cuộc tranh cãi về cách đánh giá blog, nhưng có một bạn thứ 3 xuất hiện, tham gia đánh giá làm chương trình cho điểm hoàn toàn rối loạn và ngừng hoạt động, điểm hiện nay của blog không hiển thị được. Harry và Rôn quyết định nếu một ngày nào đó đã qua điểm của blog lớn hơn m thì Harry thắng, nếu điểm blog nhỏ hơn $-m$ thì Rôn thắng. Dĩ nhiên, người nào đạt được mốc của mình trước thì người đó thắng.

Yêu cầu: Cho n, m, k, x, a_i và b_i , trong đó $a_i = 0$ nếu ngày thứ i Rôn không cho điểm và bằng 1 trong trường hợp ngược lại, $b_i = 0$ nếu ngày thứ i Harry không cho điểm và bằng 1 trong trường hợp ngược lại ($1 \leq n, m, k \leq 10^5, |x| \leq 1\,000, |x| \leq m, i = 1 \div n$). Hãy xác định số ngày tối thiểu cần thiết để Harry có thể thắng và số ngày tối thiểu cần thiết để Rôn có thể thắng. Nếu người nào không thể có khả năng thắng thì số ngày tối thiểu cần đưa ra sẽ là -1.

Dữ liệu: Vào từ file văn bản BLOG.INP:

- Dòng đầu tiên chứa các số nguyên n, m, k và x ,
- Dòng thứ i trong n dòng tiếp theo chứa 2 số nguyên a_i và b_i .

Kết quả: Đưa ra file văn bản BLOG.OUT hai số nguyên – số ngày tối thiểu cần thiết để Harry có thể thắng và số ngày tối thiểu cần thiết để Rôn có thể thắng.

Ví dụ:

| BLOG.INP | | | |
|----------|---|---|---|
| 4 | 2 | 4 | 2 |
| 1 | 0 | | |
| 1 | 0 | | |
| 1 | 0 | | |
| 1 | 0 | | |

| BLOG.OUT | |
|----------|----|
| -1 | -1 |

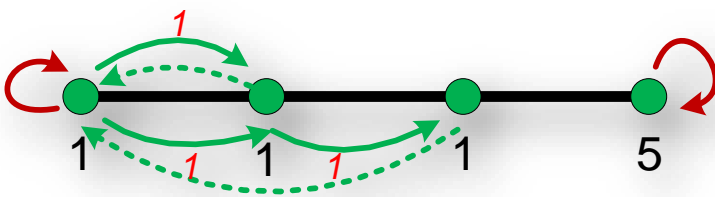


GIAO HÀNG

Tên chương trình: DELIVERY.???

Xứ Lineland có n thành phố, giữa thành phố thứ i và $i+1$ có đường nối hai chiều ($i = 1 \div n-1$). Mỗi lần vào thành phố i thì phải mua vé phí giao thông giá a_i . Tuy vậy cũng có thành phố không thu phí ($a_i = 0$), thậm chí có thành phố còn tặng thêm tiền để khuyến khích mọi người tới mua bán ($a_i < 0$).

Công ty giao hàng tại nhà có 2 trụ sở đóng ở thành phố thứ nhất và thành phố n . Có một đơn giao hàng tới n người ở n thành phố khác nhau. Mỗi chuyến đi xe chỉ chở được một mặt hàng cho một người. Việc giao hàng phải thực hiện theo nguyên tắc từ gần đến xa: chỉ được đi ngang qua các thành phố đã giao hàng. Xe của Công ty có một quyền ưu đãi: chỉ phải mua vé lượt đi, còn lúc về không phải mua vé (và cũng không nhận được tiền tặng). Xe ở trụ sở nào thì phải quay về trụ sở đó.



Yêu cầu: Cho n và a_i ($1 \leq n \leq 10^6$, $|a_i| \leq 10^4$, $i = 1 \div n$). Hãy xác định chi phí giao hàng nhỏ nhất.

Dữ liệu: Vào từ file văn bản DELIVERY.INP:

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n .

Kết quả: Đưa ra file văn bản DELIVERY.OUT một số nguyên – chi phí giao hàng nhỏ nhất.

Ví dụ:

| DELIVERY.INP |
|--------------|
| 4 |
| 1 1 1 5 |

| DELIVERY.OUT |
|--------------|
| 3 |



SIÊU XÂU CON

Tên chương trình: SUPERSUB.???

Xâu s được gọi là siêu chuỗi con của chuỗi t nếu tồn tại dãy chuỗi r_1, r_2, \dots, r_k thỏa mãn các điều kiện:

- $s = r_1 + r_2 + \dots + r_k$,
- r_i là chuỗi con các ký tự liên tiếp của t , $i = 1 \div k$.

Nếu s là siêu chuỗi con của t thì chuỗi r_1, r_2, \dots, r_k được gọi là lời giải. Lời giải tối ưu ứng với trường hợp k nhỏ nhất.

Yêu cầu: Cho chuỗi t và n chuỗi s_i , $i = 1 \div n$, độ dài của t và mỗi chuỗi s_i không vượt quá 10^6 , tổng độ dài các chuỗi không vượt quá 2×10^6 , chuỗi chỉ chứa các ký tự trong số 10 chữ cái thường đầu tiên của bảng chữ cái la tinh. Hãy xác định mỗi chuỗi s_i có phải là siêu chuỗi con của t hay không, nếu đúng – hãy chỉ ra một lời giải tối ưu. Lời giải được đưa ra dưới dạng các chuỗi con, nối với nhau bởi ký tự “|”. Nếu s_i không phải là siêu chuỗi con thì đưa ra thông báo “NO”.

Dữ liệu: Vào từ file văn bản SUPERSUB.INP:

- Dòng đầu tiên chứa chuỗi t ,
- Dòng thứ 2 chứa số nguyên n ,
- Dòng thứ i trong n dòng sau chứa chuỗi s_i .

Kết quả: Đưa ra file văn bản SUPERSUB.OUT trên n dòng, dòng thứ i lưu kết quả xử lý chuỗi s_i .

Ví dụ:

| SUPERSUB.INP |
|--------------|
| abacaba |
| 3 |
| cabab |
| dabacaba |
| aaaaaa |

| SUPERSUB.OUT |
|--------------|
| cab ab |
| NO |
| a a a a a a |



CUỘC CHIẾN

Tên chương trình: BATTLE.???

Ở một thế giới xa xôi, nơi tận cùng của vũ trụ đang có một cuộc chiến giữa Ánh sáng và Bóng tối trong n lĩnh vực. Cuộc chiến diễn ra hàng ngày và cả 2 bên đều hoàn thiện dần khả năng chiến đấu trong từng lĩnh vực.

Ban đầu cuộc chiến, ở ngày số 0 trình độ chiến đấu trong lĩnh vực i của Ánh sáng là a_i , còn của bóng tối là s_i , sau mỗi ngày trình độ của Ánh sáng tăng thêm một lượng là b_i , còn Bóng tối – tăng thêm một lượng d_i .

Một lực lượng sẽ không bị lùi bước trước lực lượng khác nếu ở mỗi lĩnh vực đều có trình độ không thua kém đối phương.

Yêu cầu: Cho n, a_i, s_i, b_i, d_i ($1 \leq n \leq 100, 0 \leq a_i, s_i, b_i, d_i \leq 1\,000, i = 1 \div n$). Hãy xác định ngày đầu tiên Ánh sáng không bị lùi bước trước Bóng tối. Nếu ngày này không xảy ra thì đưa ra thông báo “**Strong is dark side of the force.**”

Dữ liệu: Vào từ file văn bản BATTLE.INP:

- Dòng đầu tiên chứa số nguyên n ,
- Dòng thứ 2 chứa n số nguyên a_1, a_2, \dots, a_n ,
- Dòng thứ 3 chứa n số nguyên b_1, b_2, \dots, b_n ,
- Dòng thứ 4 chứa n số nguyên s_1, s_2, \dots, s_n ,
- Dòng thứ 5 chứa n số nguyên d_1, d_2, \dots, d_n .

Kết quả: Đưa ra file văn bản BATTLE.OUT một số nguyên xác định ngày đầu tiên Ánh sáng không bị lùi bước trước Bóng tối hoặc thông báo “**Strong is dark side of the force.**”

Ví dụ:

| BATTLE.INP | |
|------------|---|
| 2 | |
| 0 | 0 |
| 2 | 1 |
| 2 | 3 |
| 1 | 0 |

| BATTLE.OUT |
|------------|
| 3 |



DẪY NGOẶC

Tên chương trình: BRACKETS.???

Agnessa lần đầu tiên được làm quen với khái niệm biểu thức số học qua giờ Tin học. Cô bé quan tâm đến việc cái gì sẽ nhận được nếu ta bỏ hết các ký tự khác trong biểu thức ngoại trừ các ký tự ngoặc. Kết quả tìm kiếm trên mạng cho cô bé biết toán học gọi nó là dãy ngoặc và cô bé còn biết thêm thế nào là dãy ngoặc đúng.

Ví dụ $() (())$ là dãy ngoặc đúng vì nó có thể nhận được từ một biểu thức số học, chẳng hạn $(2+2):(3-(5-2)+4)$, còn các dãy ngoặc $(())$ hoặc $(())$ là không đúng. Dễ dàng thấy rằng, với 6 dấu ngoặc mở và đóng, trong đó có 3 ngoặc mở và 3 ngoặc đóng chỉ tồn tại 5 dãy ngoặc đúng: $((()))$, $((())())$, $(())(())$, $(())(())$ và $(())(())$.

Agnessa thích thú tìm hiểu các phép biến đổi biểu thức ngoặc đúng, bắt đầu từ việc thêm ngoặc. Cô bé mau chóng nhận ra rằng nếu thêm một ngoặc thì dãy ngoặc không còn đúng, còn nếu thêm 2 ngoặc thì có thể có số dãy vẫn đúng. Ví dụ, với dãy ngoặc đúng $()()$, nếu thêm 2 dấu ngoặc ta có thể nhận được các dãy ngoặc đúng $((()))$, $((())())$, $(())(())$, và $(())(())$. Dễ dàng nhận thấy rằng chỉ có thể nhận được dãy ngoặc đúng nếu thêm một dấu ngoặc mở và một dấu ngoặc đóng. Ví dụ, từ dãy ngoặc đúng đơn giản nhất $()$ ta có 7 cách thêm 2 dấu ngoặc để có dãy ngoặc đúng mới $((()))$, $((())())$, $(())(())$, $(())(())$, $(())(())$, $(())(())$, $(())(())$. Các dấu ngoặc thêm vào được in đậm.

Trong dãy ngoặc mới dấu mở ngoặc mới nằm ở vị trí i và dấu ngoặc đóng nằm ở vị trí j . Hai cách thêm ứng với các cặp (i_1, j_1) và (i_2, j_2) gọi là khác nhau nếu $i_1 \neq i_2$ hoặc $j_1 \neq j_2$.

Yêu cầu: Cho dãy ngoặc đúng độ dài $2n$ ($1 \leq n \leq 50\,000$). Hãy xác định số lượng cách thêm khác nhau 2 ngoặc để nhận được dãy ngoặc đúng độ dài $2n+2$.

Dữ liệu: Vào từ file văn bản BRACKETS.INP gồm một dòng chứa dãy ngoặc đúng độ dài $2n$.

Kết quả: Đưa ra file văn bản BRACKETS.OUT một số nguyên – số lượng cách thêm khác nhau 2 ngoặc để nhận được dãy ngoặc đúng độ dài $2n+2$.

Ví dụ:

| |
|--------------|
| BRACKETS.INP |
| () |

| |
|--------------|
| BRACKETS.OUT |
| 7 |



TRƯỜNG ĐÀO TẠO

Tên chương trình: SCHOOL.???

Để chuẩn bị cho World Cup 2018 người ta tổ chức một trường đào tạo để chuẩn bị nguồn cho đội tuyển Quốc Gia. Trường sẽ huấn luyện cho m thanh niên sinh các năm từ 1994 đến 1996. Có n người nộp đơn dự tuyển. Sau khi thực hiện các bài thi mỗi người có một điểm số đánh giá năng khiếu và trình độ chuyên môn. Điểm số của mỗi người khác nhau từng đôi một. Nhà trường dự định chọn a người sinh năm 1994, b người sinh năm 1995 và c người sinh năm 1996 ($a+b+c=m$). Ngoài ra còn phải thỏa mãn điều kiện: điểm nhỏ nhất của người sinh năm 1994 phải lớn hơn điểm nhỏ nhất của người sinh năm 1995, điểm nhỏ nhất của người sinh năm 1995 phải lớn hơn điểm nhỏ nhất của người sinh năm 1996, tất cả những người cùng năm sinh có điểm lớn hơn điểm nhỏ nhất của người cùng năm sinh được chọn sẽ phải được chọn.

Yêu cầu: Cho a, b, c, n và và điểm của mỗi người dự thi tuyển. Gọi số người được tuyển sinh năm 1994 là m_{94} , số người được tuyển sinh năm 1995 là m_{95} , số người được tuyển sinh năm 1996 là m_{96} ($m_{94}+m_{95}+m_{96} = m$). Hãy xác định số người được chọn sao cho $F = |m_{94}-a|+|m_{95}-b|+|m_{96}-c|$ là nhỏ nhất. Mỗi lứa tuổi phải có ít nhất một người được chọn.

Dữ liệu: Vào từ file văn bản SCHOOL.INP:

- Dòng đầu tiên chứa số nguyên t – số lượng tests
- Mỗi test cho trên một nhóm dòng:
 - Dòng đầu tiên 3 số nguyên a, b, c ($a+b+c \leq n$),
 - Dòng thứ 2 chứa số nguyên n , tổng các n trong t tests không vượt quá 300 000,
 - Mỗi dòng trong n dòng sau chứa 2 số năm sinh và số điểm của mỗi người (xem ví dụ), điểm là số nguyên trong phạm vi từ 1 đến 10^9 .

Kết quả: Đưa ra file văn bản SCHOOL.OUT, nếu không có cách chọn thì đưa ra số -1, nếu có cách chọn thì đưa ra F và các giá trị m_{94}, m_{95}, m_{96} . Nếu tồn tại nhiều cách chọn khác nhau thì đưa ra một cách chọn bất kỳ.

Ví dụ:

| SCHOOL.INP | SCHOOL.OUT |
|------------|------------|
| 1 | 2 3 2 1 |
| 2 3 1 | |
| 7 | |
| 1996 2 | |
| 1994 7 | |
| 1994 4 | |
| 1996 1 | |
| 1995 3 | |
| 1994 5 | |
| 1995 6 | |

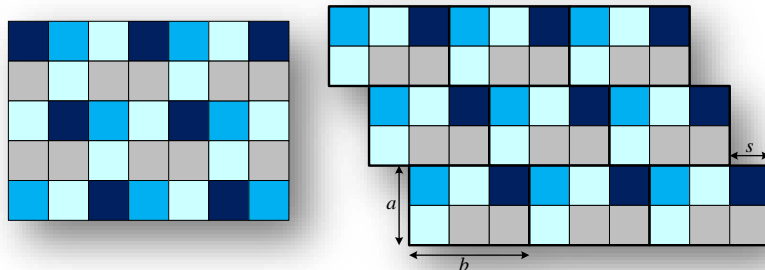


TÀU CON THOI

Tên chương trình: SHUTLLE.???

Để kỷ niệm mốc 50 năm con người bay vào vũ trụ người ta quyết định xây dựng một tàu con thoi thế hệ mới. Phần diện tích hình chữ nhật của vỏ tàu (ta sẽ gọi ngắn gọn là hình chữ nhật) phải được lát bằng các miếng chịu nhiệt hình vuông cùng kích thước. Có k loại miếng, mỗi loại đặc trưng bằng một ký tự la tinh in hoa trong số k chữ cái đầu tiên của bảng chữ cái la tinh.

Việc gắn trực tiếp từng miếng chịu nhiệt lên vỏ tàu là rất khó khăn, vì vậy người ta gắn chúng thành các tấm hình chữ nhật kích thước $a \times b$, sau đó gắn các tấm này vào hình chữ nhật theo từng hàng, từ trên xuống dưới. Các tấm phải được gắn theo cùng một chiều, nghĩa là nếu ta tịnh tiến song song một tấm cho trùng với tấm khác thì màu các miếng trên 2 tấm đó phải trùng nhau. Các tấm có thể nhô ra khỏi biên của hình chữ nhật. Khi gắn, hàng dưới có thể được tịnh tiến so với hàng trên một khoảng là s sang phải ($0 \leq s < b$).



Kỹ sư trưởng muốn chuẩn bị các tấm sao cho diện tích của tấm là nhỏ nhất.

Yêu cầu: Cho r, c – kích thước hình chữ nhật (theo đơn vị miếng chịu nhiệt), bản đồ các loại miếng chịu nhiệt cần gắn dưới dạng r xâu độ dài c chỉ chứa các ký tự in hoa đầu tiên của bảng chữ cái latin. Hãy xác định a, b và r tương ứng với tấm có diện tích nhỏ nhất. Dữ liệu đảm bảo tồn tại a và b thỏa mãn $2a \leq r$ và $2b \leq c$.

Dữ liệu: Vào từ file văn bản SHUTLLE.INP:

- Dòng đầu tiên chứa 2 số nguyên r và c ($2 \leq r, c \leq 1961$),
- Mỗi dòng trong r dòng sau chứa xâu độ dài c mô tả một dòng của hình chữ nhật.

Kết quả: Đưa ra file văn bản SHUTLLE.OUT trên một dòng 3 số nguyên a, b và s .

Ví dụ:

| SHUTLLE.INP |
|-------------|
| 5 7 |
| DCADCAD |
| BABBABB |
| ADCADCA |
| BBABBAB |
| CADCADC |

| SHUTLLE.OUT |
|-------------|
| 2 3 1 |



CÔNG VIÊN GIẢI TRÍ

Tên chương trình: ATTRACT.???

Ô tô buýt chở đoàn n học sinh tham dự Olympic Tin học tới công viên giải trí của thành phố vào thời điểm 0. Trong công viên có gian phòng trò chơi với m máy. Ở mỗi thời điểm mỗi máy chỉ dành cho một người chơi. Trò chơi ở máy thứ i kéo dài t_i đơn vị thời gian và người chơi không được phép ngắt giữa chừng.

Ban Tổ chức phải bố trí lịch chơi sao cho mỗi học sinh được chơi trên tất cả các máy và sau đó chở học sinh về được sớm nhất. Thời gian chuyển từ ô tô buýt vào phòng chơi và thời gian chuyển từ máy này sang máy khác là không đáng kể. Tại mỗi thời điểm một học sinh có thể chơi trên máy hoặc đi dạo công viên, chờ đợi đến lượt mình.

Yêu cầu: Cho n , m và t_i ($1 \leq m \leq n \leq 100$, $1 \leq t_i \leq 100$, $i = 1 \div m$). Hãy chỉ ra thời điểm ô tô buýt khởi hành ra về và với mỗi học sinh – chỉ ra thời điểm được bắt đầu chơi trên máy thứ i ($i = 1 \div m$).

Dữ liệu: Vào từ file văn bản ATTRACT.INP:

- Dòng đầu tiên chứa 2 số nguyên n và m ,
- Dòng thứ 2 chứa m số nguyên t_1, t_2, \dots, t_m .

Kết quả: Đưa ra file văn bản ATTRACT.OUT: dòng đầu tiên chứa một số nguyên – thời điểm ô tô buýt khởi hành ra về, tiếp theo là n nhóm dòng, nhóm thứ j tương ứng với học sinh j , mỗi nhóm bắt đầu bằng một dòng trống, sau đó m dòng, mỗi dòng chứa 2 số nguyên k và v cho biết học sinh này bắt đầu chơi trên máy k từ thời điểm v , các dòng trong nhóm được sắp xếp theo thứ tự tăng dần của v .

Ví dụ:

| ATTRACT.INP | |
|-------------|---|
| 3 | 2 |
| 2 | 1 |



| ATTRACT.OUT | |
|-------------|---|
| 6 | |
| 1 | 0 |
| 2 | 2 |
| 1 | 2 |
| 2 | 4 |
| 2 | 0 |
| 1 | 4 |