

ĐỀ CHÍNH THỨC
KHÔNG THI

Thời gian: 180 phút, không kể thời gian phát đề.

(Đề thi gồm 04 trang)

TỔNG QUAN VỀ ĐỀ THI

| | File nguồn nội | File dữ liệu | File kết quả | Thời gian mỗi test | Biểu điểm |
|-------|-------------------|--------------|--------------|-----------------------|--------------|
| Bài 1 | PRIZES.* | PRIZES.INP | PRIZES.OUT | 1 giây | 6 điểm |
| Bài 2 | RIVER.* | RIVER.INP | RIVER.OUT | 1 giây | 7 điểm |
| Bài 3 | SEARCH.* | SEARCH.INP | SEARCH.OUT | 1 giây | 7 điểm |

(Phần mở rộng * là PAS hay CPP tùy theo ngôn ngữ và môi trường lập trình Free Pascal hay Dev C++)

Bài 1: Alice và Bob trúng thưởng online và bây giờ phải chọn phần thưởng. Có tổng cộng n phần thưởng đánh số từ 1 đến n .

Phần thưởng được phát như sau: nhà tổ chức sẽ thông báo cho người thắng cuộc số nguyên dương k ($1 \leq k \leq n/3$). Đầu tiên Alice chọn cho mình k phần thưởng có số thứ tự liên tiếp ở vị trí bất kì. Sau đó Bob chọn cho mình k phần thưởng có số thứ tự liên tiếp mà không trùng với phần thưởng của Alice. Sau đó là cả hai sẽ được nhận thưởng.

Alice biết rằng đối với Bob các phần thưởng có giá trị khác nhau (giá trị là một số nguyên dương). Vì vậy Bob nên Alice muốn chọn phần thưởng sao cho tổng giá trị giải thưởng của Bob càng bé càng tốt. Đối với Alice thì giải thưởng nào cũng như nhau.

Cho giá trị giải thưởng và số k , hãy viết chương trình tìm giá trị x nhỏ nhất mà Alice có thể nhận thưởng để tổng giá trị giải thưởng của Bob không thể lớn hơn x .

Dữ liệu vào từ file văn bản **PRIZES.INP**:

- Dòng đầu tiên chứa 2 số nguyên: n — số lượng phần thưởng và k — số lượng phần thưởng mỗi bạn có thể nhận ($3 \leq n \leq 100.000$, $1 \leq k \leq n/3$).
- Dòng thứ 2 chứa n số nguyên dương: a_1, a_2, \dots, a_n là giá trị của phần thưởng tương ứng đối với Bob ($1 \leq a_i \leq 10^9$).

Kết quả ghi ra file văn bản **PRIZES.OUT** chứa một số duy nhất — giá trị x nhỏ nhất cần tìm.

Ví dụ:

| PRIZES.INP | PRIZES.OUT | |
|-----------------------------|------------|---|
| 10 2 1 2 4 5 2 4 2 2 1 6 | 7 | Alice có thể chọn phần thưởng số 4 và 5. Sau đó Bob chỉ có thể chọn phần thưởng 9 và 10 với tổng giá trị là 7 |

Chú ý:

- 30% số test $3 \leq n \leq 50, 1 \leq a_i \leq 10^5$
- 30% số test $3 \leq n \leq 5000, 1 \leq a_i \leq 10^5$
- 40% số test $3 \leq n \leq 100\,000, 1 \leq a_i \leq 10^9$

Bài 2: Ở một đất nước nọ có 1 dòng sông lớn. Nhiều năm trước dòng sông được phân chia cho n công ty đánh cá được đánh số từ 1 đến n tính từ đầu nguồn. Công ty i quản lý đoạn sông độ dài a_i .

Từ đó đến nay ở nước đó đã xảy ra k sự kiện. Mỗi sự kiện chỉ thuộc 1 trong 2 loại sau:

hoặc là một công ty phá sản, hoặc là công ty nào đó tách ra làm 2 công ty.

Khi có sự kiện xảy ra với 1 công ty thì đoạn sông công ty đó quản lý sẽ được chia ra như sau: độ dài đoạn sông phải lớn hơn hoặc bằng 2, nếu độ dài đoạn sông là số chẵn thì chia đôi, còn nếu là số lẻ thì chia ra làm 2 đoạn sông mới mà đoạn ở gần đầu nguồn sẽ ngắn hơn đoạn còn lại là 1 đơn vị.

Nếu một công ty bị phá sản thì đoạn sông mà công ty đó quản lý sẽ được chia ra cho các công ty hàng xóm. Nếu chỉ có một hàng xóm thì tiếp quản cả đoạn sông. Nếu có 2 hàng xóm thì chia theo nguyên tắc trên.

Nếu một công ty mà tách ra làm 2 công ty mới thì đoạn sông được chia theo đúng nguyên tắc trên.

Theo cách đó thì sau mỗi một sự kiện mỗi công ty vẫn quản lý một đoạn sông.

Quản lý sông thì phải nộp thuế, thuế công ty phải nộp bằng bình phương độ dài đoạn sông công ty đó quản lý. Để kiểm kê thuế, nhà quản lý cần có dữ liệu tổng số thuế sau mỗi lần thay đổi.

Hãy viết chương trình tính tổng số thuế ban đầu và sau mỗi một sự kiện.

Dữ liệu vào từ file văn bản **RIVER.INP**

- Dòng đầu chứa 2 số nguyên: n và p — số lượng công ty ($2 \leq n \leq 100\,000$) và mức độ khó của test ($0 \leq p \leq 4$).
- Dòng 2 chứa n số nguyên a_1, a_2, \dots, a_n — độ dài các đoạn sông.
- Dòng 3 chứa số nguyên k — số lượng sự kiện ($1 \leq k \leq 100\,000$).
- k dòng tiếp theo chứa thông tin các sự kiện: dòng thứ i chứa 2 số nguyên: e_i và

v_i — loại sự kiện và công ty mà sự kiện xảy ra. Nếu $e_i = 1$ thì tức là công ty v_i tại thời điểm hiện tại phá sản, còn nếu $e_i = 2$ thì tức là công ty tách ra làm 2.

- Đảm bảo rằng giá trị v_i không vượt quá số lượng công ty tại thời điểm hiện tại. Đoạn sông bị chia có độ dài lớn hơn hoặc bằng 2. Nếu cả dòng sông chỉ còn 1 công ty thì nó ko bị phá sản.

Kết quả ghi ra file văn bản **RIVER.OUT** chứa $(k + 1)$ dòng, trên mỗi dòng viết 1 số nguyên. Dòng đầu tiên là tổng số thuế lúc đầu. k dòng tiếp theo là tổng số thuế sau mỗi sự kiện tương ứng.

Ví dụ:

| RIVER.INP | RIVER.OUT |
|-----------|-----------|
| 4 0 | 75 |
| 3 5 5 4 | 105 |
| 5 | 73 |
| 1 1 | 101 |
| 2 1 | 83 |
| 1 3 | 113 |
| 2 2 | |
| 1 3 | |

Chú ý:

- 30% số test $2 \leq n \leq 100, 1 \leq k \leq 100, 1 \leq a_i \leq 100, p = 1$
- 30% số test $2 \leq n \leq 100\ 000, 1 \leq k \leq 100\ 000, 1 \leq a_i \leq 10^4, p = 2$. Đối với i từ 1 đến k
 - 1 thỏa mãn điều kiện: $|v_i - v_{i+1}| \leq 10$
- 20% số test $2 \leq n \leq 100\ 000, 1 \leq k \leq 100\ 000, 1 \leq a_i \leq 10^4, p = 3$. Đối với tất cả các sự kiện $e_i = 1$ (tức là chỉ phá sản).
- 20% số test $2 \leq n \leq 100\ 000, 1 \leq k \leq 100\ 000, 1 \leq a_i \leq 10^4, p = 4$

Bài 3:

Để tổ chức giải vô địch tìm kiếm trên mạng Meganet nhà tổ chức cần phải hạn chế địa chỉ tìm kiếm. Địa chỉ trên mạng Meganet là một chuỗi kí tự bao gồm tên server và tên trang web.

Tên server là một chuỗi kí tự chứa từ 1 đến 5 phần. Mỗi phần là 1 chuỗi không rỗng chứa các chữ cái latin không viết hoa. Các phần phân biệt bằng dấu chấm. Ví dụ tên server:

«a», «ab.cd», «abacaba», «a.b.c.d.e».

Tên site là một chuỗi kí tự có thể rỗng hoặc có thể chứa từ 1 đến 5 phần. Mỗi phần bắt đầu từ kí tự «/», sau đó là các kí tự latin không viết hoa. Ví dụ tên site: «», «/a», «/aba», «/a/b/c/d/e».

Địa chỉ bao gồm tên server, sau đó là tên site. Ví dụ: «a», «aba/d/f/g/h», «a.b», «aba.caba/def/g», «c.d.e.f.g/a/b/c/d/e».

Để hạn chế truy cập đến các địa chỉ mạng Meganet nhà tổ chức phải chuẩn bị một số filter. Filter giống địa chỉ cũng gồm 2 phần: filter server và filter site.

Filter server cũng giống tên server nhưng trước đó có thể có kí tự «*». Nếu filter server chỉ gồm tên server thì nó chỉ tương ứng với những server cùng tên. Còn nếu filter server có dạng «*.S», trong đó S — tên server, thì nó tương ứng với các server mà khi xóa đi phần đầu thì còn lại xâu S.

Tương tự, *filter site* chứa tên site và sau đó có thể có kí tự «/*». Filter site chỉ chứa tên site thì tương ứng với những site trùng tên hoàn toàn. Nếu filter site có dạng «R/*» thì nó tương ứng với những site mà khi xóa đi phần cuối trong tên ta thu được xâu R.

Địa chỉ *phù hợp* với filter khi mà tên server phù hợp với filter sever và đồng thời tên site phù hợp với filter site.

Ví dụ về filters và địa chỉ tương ứng:

| Filter | Ví dụ các địa chỉ phù hợp | | |
|-----------|---------------------------|------------------|---------------------|
| ab.c/d/e | ab.c/d/e | | |
| *.a | a | ax.a | efg.a |
| *.a/b/c | a/b/c | x.a/b/c | e.fg.a/b/c |
| x.yz/a/* | x.yz/a | x.yz/a/b/c | x.yz/a/xyz |
| *.a/* | a | x.a | e.fg.a |
| | a/b/c | x.a/ddd/c | e.fg.a/b/c/g/haha/i |
| *.a/b/c/* | a/b/c | x.a/b/c | e.fg.a/b/c |
| | a/b/c/xxx | e.fg.a/b/c/d/e/f | |

Yêu cầu: Cho danh sách filter và danh sách địa chỉ, hãy viết chương trình tính số lượng filter phù hợp với từng địa chỉ.

Dữ liệu vào từ file văn bản **SEARCH.INP**

- Dòng đầu chứa 2 số nguyên: n — số lượng filter ($1 \leq n \leq 50.000$) và p — mức độ khó ($0 \leq p \leq 3$).
- n dòng tiếp theo chứa filter theo đúng mô tả trên.
- Dòng tiếp theo cho số nguyên k — số lượng địa chỉ cần kiểm tra ($1 \leq k \leq 50.000$).
- k dòng tiếp theo chứa các địa chỉ theo đúng quy tắc đặt ra.
- Độ dài mỗi xâu không quá 50 kí tự.
- Tổng độ lớn file input không quá 4 MB.

Kết quả ghi ra file văn bản **SEARCH.OUT** chứa k số nguyên, trên mỗi dòng — số lượng filter phù hợp với địa chỉ tương ứng.

Ví dụ:

| SEARCH.INP | SEARCH.OUT |
|------------|------------|
| 2 0 | 0 |
| a.bb/c | 1 |
| bb/c/d | 0 |
| 4 | 0 |
| a.bb | |
| bb/c/d | |
| a.bb/c/d | |
| bb/c | |
| 4 0 | 0 |
| *.bb/c | 4 |
| *.bb/c/* | 3 |
| bb/c | 0 |
| bb/c/* | 2 |
| 6 | 1 |
| bb bb/c | |
| bb/c/d | |
| a.bb | |
| a.bb/c | |
| a.bb/c/d | |

Chú ý:

- 27% số test $1 \leq n \leq 1000, 1 \leq k \leq 1000, p = 1$. Filter bắt đầu bằng «*.» và kết thúc bằng «/*».
- 25% số test $1 \leq n \leq 50\,000, 1 \leq k \leq 50\,000, p = 2$. Filter không chứa kí tự «*».
- 48% số test $1 \leq n \leq 50\,000, 1 \leq k \leq 50\,000, p = 3$

----Hết----