

Bảng B

Bài 1. Đóng gói sản phẩm

Tên file chương trình: ZXY.PAS

Ở đầu ra của một dây chuyền sản xuất trong nhà máy ZXY có một máy xếp tự động. Sau khi kết thúc việc gia công trên dây chuyền, các sản phẩm sẽ được xếp vào các hộp có cùng dung lượng M . Sản phẩm rời khỏi dây chuyền được xếp vào hộp đang mở (khi bắt đầu ca làm việc có một hộp rỗng được mở sẵn) nếu như dung lượng của hộp còn đủ để chứa sản phẩm. Trong trường hợp ngược lại, máy sẽ tự động đóng nắp hộp hiện tại, cho xuất xưởng rồi mở một hộp rỗng mới để xếp sản phẩm vào. Trong một ca làm việc có n sản phẩm đánh số từ 1 đến n theo đúng thứ tự mà chúng rời khỏi dây chuyền. Sản phẩm thứ i có trọng lượng là a_i , $i = 1, 2, \dots, n$. Ban Giám đốc nhà máy qui định rằng sản phẩm xuất xưởng của mỗi ca làm việc phải được xếp vào trong không quá k hộp.

Yêu cầu: Hãy giúp người quản đốc của ca làm việc xác định giá trị M nhỏ nhất sao cho số hộp mà máy tự động cần sử dụng để xếp đầy n sản phẩm xuất xưởng của ca không vượt quá số k cho trước.

Dữ liệu: Vào từ file văn bản ZXY.INP:

- Dòng đầu tiên chứa hai số nguyên n và k , ($1 \leq k \leq n \leq 15000$);
- Dòng thứ i trong n dòng tiếp theo chứa số nguyên dương a_i ($a_i \leq 30000$), $i = 1, 2, \dots, n$.

Các số trên một dòng cách nhau ít nhất một dấu cách.

Kết quả: Ghi ra file ZXY.OUT một số nguyên duy nhất là dung lượng của hộp.

Ví dụ:

ZXY . INP	ZXY . OUT
9 4	5
1	
1	
1	
3	
2	
2	
1	
3	
1	

Bài 2. Chữ số**Tên chương trình: DIGIT.PAS**

Cho chuỗi M không quá 127 ký tự lấy từ tập $F = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$ và không bắt đầu bằng ký tự 0. Gọi S là chuỗi với giá trị ban đầu là chuỗi M .

Người ta biến đổi M theo quy tắc sau: đếm số lần xuất hiện các ký tự $0, 1, 2, \dots, F$, gọi K_i là số lần xuất hiện ký tự i (với i lần lượt là $0, 1, 2, \dots, F$). Với các $K_i \neq 0$ người ta viết liên tiếp chuỗi biểu diễn số K_i trong cơ số 16 và ký tự i . Chuỗi kết quả thu được là giá trị mới của M . Sau mỗi lần biến đổi người ta lại viết tiếp M vào sau S .

Ví dụ, với $M = '150A'$, S nhận giá trị ban đầu là $'150A'$.

Sau lần biến đổi thứ nhất ta có M là $'1011151A'$ và $S = '150A1011151A'$.

Sau lần biến đổi thứ 2 ta có M là $'1051151A'$ và $S = '150A1011151A1051151A'$.

Sau lần biến đổi thứ 3 ta có M là $'1041251A'$ và $S = '150A1011151A1051151A1041251A'$.

Yêu cầu: Cho chuỗi M , số lần biến đổi L ($0 \leq L \leq 10^7$) và X là một ký tự từ tập F . Hãy đếm số lần xuất hiện X trong S thu được sau L lần biến đổi M .

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

- Dòng thứ nhất chứa chuỗi M ,
- Dòng thứ 2 chứa số nguyên L
- Dòng thứ 3 chứa ký tự X .

Kết quả: Đưa ra file văn bản DIGIT.OUT một số nguyên - số lần xuất hiện X .

Ví dụ:

DIGIT . INP
150A
3
2

DIGIT . OUT
1

Bài 3. Đổi đất**Tên file chương trình: LAND.PAS**

Hai bộ lạc Anpha và Bêta sống rất hoà thuận với nhau. Một phần ranh giới của hai bộ lạc là một đường gấp khúc không tự cắt. Đường gấp khúc nhận được bằng cách lần lượt nối N điểm đôi một khác nhau A_1, A_2, \dots, A_N . Điểm A_i được xác định bởi hoành độ x_i và tung độ y_i (x_i là các số nguyên thoả mãn điều kiện: $x_i \leq x_{i+1}$). Phần đất của bộ lạc Anpha nằm ở phía trên đường gấp khúc.

Nhân dịp năm mới, tù trưởng hai bộ lạc quyết định thay đường ranh giới cũ bằng cách xây dựng một đường cao tốc nối thẳng từ A_1 tới A_N và lấy đường cao tốc này ranh giới mới. Dĩ nhiên, sự thay đổi này sẽ chuyển một phần đất của bộ lạc Anpha cho bộ lạc Bêta và ngược lại. trưởng thoả thuận phần diện tích chênh lệch do việc thay đường ranh giới sẽ được điều chỉnh trong tương lai bằng cách khác.

Yêu cầu: Hãy tính diện tích phần đất SA của bộ lạc trở thành đất của bộ lạc Bêta và diện tích phần đất SB của bộ lạc Bêta trở thành đất của bộ lạc Anpha sau thay đổi đường ranh giới giữa hai bộ lạc.

Dữ liệu: Vào từ file văn bản LAND.INP trong đó :

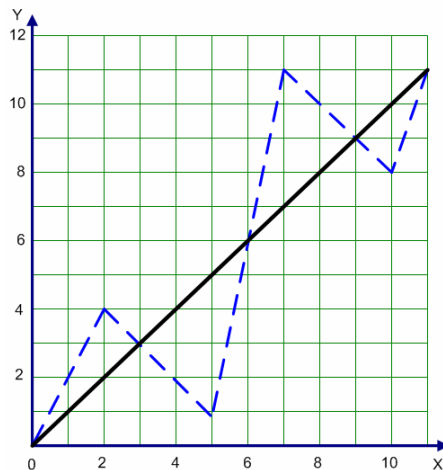
- Dòng đầu chứa số N ($N \leq 10000$) ;
- Dòng thứ i trong N dòng tiếp theo chứa hai số nguyên x_i và y_i đặt cách nhau ít nhất 1 dấu cách ($-32000 \leq x_i, y_i \leq 32000$).

Kết quả: Đưa ra file văn bản LAND.OUT trong đó dòng thứ nhất chứa SA , dòng thứ hai chứa SB . Kết quả được lấy chính xác với 4 chữ số sau dấu chấm thập phân.

Ví dụ :

LAND . INP	
6	
0	0
2	4
5	1
7	11
10	8
11	11

LAND . OUT	
8.0000	
9.0000	



đổi
tốc là
làm
số
Hai từ
đổi
một

Anpha
khi

Bài 4. Bộ sưu tập**Tên chương trình: COLLECT.PAS**

Một bộ sưu tập tiền xu cổ được coi là có giá trị phải gồm không ít hơn Z_0 đồng tiền vàng, S_0 đồng tiền bạc và M_0 đồng tiền đồng. Bộ sưu tập ban đầu của Alibaba có một số lượng nhất định các đồng tiền vàng, bạc và đồng nhưng chưa phải là một bộ sưu tập có giá trị. Tại Trụ sở của Hiệp hội những người sưu tầm tiền cổ có đặt một máy đổi tiền để giúp hội viên đổi được các bộ sưu tập có giá trị. Tuy nhiên, máy đổi chỉ hỗ trợ việc đổi tiền trọn gói theo quy tắc đổi gói (Z_1, S_1, M_1) lấy gói (Z_2, S_2, M_2) đồng tiền. Các quy tắc đổi tiền khác nhau từng đôi một, được gán số hiệu tuần tự 1, 2, 3, ... và được công bố trước. Hội viên có thể tạo gói tiền thích hợp từ bộ sưu tập của mình để thực hiện việc đổi tiền. Số lần đổi tiền là không hạn chế, tuy nhiên, để ngăn chặn việc đầu cơ, Hiệp hội quy định mỗi hội viên không được giữ quá 4 đồng tiền mỗi loại. Các đồng tiền nhận được sau mỗi lần đổi được gộp lại với các đồng tiền mà hội viên đang có để thành *một bộ sưu tập mới* và có thể được sử dụng để đổi trong những lần sau nếu cần.

Yêu cầu: Cho biết số lượng Z, S, M các đồng tiền vàng, bạc, đồng mà Alibaba có ban đầu và các quy tắc đổi tiền. Hãy chỉ ra *một phương án đổi tiền* nào đó để Alibaba có được bộ sưu tập có giá trị. Dữ liệu vào đảm bảo luôn có phương án.

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

- Dòng đầu ghi 6 số nguyên không âm Z, S, M, Z_0, S_0, M_0 ($0 \leq Z, S, M, Z_0, S_0, M_0 \leq 4$);
- Các dòng tiếp theo mỗi dòng ghi 6 số nguyên không âm $Z_1 S_1 M_1 Z_2 S_2 M_2$ xác định một quy tắc đổi tiền.

Kết quả: Đưa ra file văn bản COLLECT.OUT một dòng ghi dãy số hiệu các quy tắc theo thứ tự đã sử dụng trong phương án đổi tiền.

Các số trên một dòng của file dữ liệu và file kết quả đặt cách nhau ít nhất một dấu cách.

Ví dụ:

COLLECT . INP					
4	0	1	3	3	3
1	0	1	0	2	2
0	1	1	0	0	3
2	0	1	1	2	3
1	0	0	1	1	0

COLLECT . OUT	
3	4