

Sử dụng ngôn ngữ lập trình Pascal để lập trình giải các bài toán sau:

Câu 1(3.0 điểm): Giải nén xâu

UNZIP.PAS

Trong máy tính, để tiết kiệm bộ nhớ, người ta thường tìm cách nén dữ liệu. Trong việc nén dữ liệu dạng văn bản, ta sử dụng một phương pháp đơn giản được mô tả thông qua ví dụ sau: với xâu ký tự: 'aaaabbbb' sẽ được nén lại thành xâu '4a3b'.

Cho một xâu ký tự St1 gồm các ký tự thuộc tập 'a'..'z'. Gọi St là xâu nén của xâu St1 theo phương pháp được mô tả như trên. Xâu St gồm N ($1 \leq N \leq 255$) ký tự thuộc tập các ký tự: 'a'..'z', '0'..'9'

Yêu cầu: Hãy giải nén xâu St để được xâu gốc St1.

Dữ liệu vào: Cho trong file văn bản UNZIP.INP có cấu trúc như sau:

- Dòng 1: Ghi xâu ký tự St.

Dữ liệu ra: Ghi ra file văn bản UNZIP.OUT theo cấu trúc như sau:

- Dòng 1: Ghi xâu St1 là xâu sau khi đã được giải nén.

Ví dụ:

UNZIP.INP	UNZIP.OUT
12a5bk2c	aaaaaaaaaaaabbbbbbkc

Câu 2 (3.0 điểm): Đường chạy địa hình

ROUTE.PAS

Trong Đại hội thể thao Quốc tế, người ta dự định sẽ tổ chức một môn chạy bộ địa hình. Đường chạy địa hình là một đường khép kín, điểm bắt đầu cũng là điểm kết thúc. Đường chạy có độ dài N (mét), mỗi mét có một độ cao h (cm) so với mực nước biển.

Yêu cầu: Hãy đếm số lượng đường bằng, số lượng đường dốc lên và số lượng đường dốc xuống của đường chạy địa hình, tính từ điểm xuất phát.

Dữ liệu vào: Cho trong file văn bản ROUTE.INP có cấu trúc như sau:

- Dòng 1: Ghi số nguyên dương N , là chiều dài của đường chạy địa hình.

- Dòng 2: Ghi N số nguyên dương h_i là độ cao của mét thứ i trên đường chạy địa hình. Các số được ghi cách nhau ít nhất một dấu cách ($3 \leq N \leq 30000$; $1 \leq h_i \leq 30000$).

Dữ liệu ra: Ghi ra file văn bản ROUTE.OUT theo cấu trúc như sau:

- Dòng 1: Ghi ba số nguyên dương x y z , trong đó x là số lượng đoạn đường bằng, y là số lượng đoạn đường dốc lên, z là số lượng đoạn đường dốc xuống của đường chạy địa hình. Các số được ghi cách nhau ít nhất một dấu cách.

Ví dụ:

ROUTE.INP	ROUTE.OUT
6	2 1 2
20 23 60 50 50 20	

Câu 3 (2.0 điểm): Chuỗi kết nối

CONNECT.PAS

Cho một tập gồm M bộ ba các số nguyên dương và hai số nguyên dương A, B . Một chuỗi kết nối hai số A, B là dãy các bộ ba có dạng: $A_0 W_0 A_1, A_1 W_1 A_2, A_2 W_2 A_3, \dots, A_{k-2} W_{k-2} A_{k-1}, A_{k-1} W_{k-1} A_k$. Thỏa mãn các điều kiện sau:

1. $A_0 = A$
2. $A_k = B$
3. Với mỗi $i, 0 < i < k$, hoặc A_i, W_i, A_{i+1} hoặc là A_{i+1}, W_i, A_i phải là các số bộ ba đã cho.

Nếu tồn tại một chuỗi kết nối như vậy thì giá trị của chuỗi được tính bằng cách cộng các số W_i xuất hiện trong chuỗi.

Yêu cầu: Tìm chuỗi kết nối hai số A, B có giá trị bé nhất.

Dữ liệu vào: Ghi trong tệp văn bản CONNECT.INP có cấu trúc như sau:

- Dòng 1: Ghi số M ($M \leq 4 \times 10^6$) và hai số A, B ($1 \leq A, B \leq 32000$).
- M dòng tiếp theo: Mỗi dòng ghi bộ ba $X_i Y_i Z_i$ ($1 \leq X_i, Y_i, Z_i \leq 32000$) các số nguyên dương.

Dữ liệu ra: Ghi ra file văn bản CONNECT.OUT theo cấu trúc như sau:

- Dòng 1: Ghi chữ YES hoặc NO cho biết có tồn tại chuỗi kết nối hai số A, B hay không.
- Dòng 2: Nếu dòng đầu ghi YES thì dòng này chứa giá trị của chuỗi kết nối hai số A, B có giá trị bé nhất. Nếu dòng đầu ghi NO thì dòng này không tồn tại.

Ví dụ:

CONNECT.INP	CONNECT.OUT
9 1 11	YES
1 2 5	3
5 2 6	
1 3 8	
8 1 11	
1 1 6	
10 1 6	
11 3 6	
11 4 8	
10 1 11	

Câu 4 (2.0 điểm): Dãy chia hết**SIGN.PAS**

Xét một dãy số nguyên A gồm N phần tử a_1, a_2, \dots, a_N . Viết dãy số đó theo thứ tự từ trái sang phải, sau đó đặt giữa mỗi cặp số cạnh nhau dấu cộng '+' hoặc trừ '-', khi đó ta thu được một biểu thức số học. Ta nói dãy số là chia hết cho K nếu tồn tại một cách đặt dấu để thu được biểu thức số học chia hết cho K .

Yêu cầu: Hãy xác định dãy số đã cho có chia hết cho K hay không.

Dữ liệu vào: Cho trong file văn bản SIGN.INP có cấu trúc như sau:

- Dòng 1: Ghi hai số nguyên N và K ($2 \leq N \leq 10^4$, $2 \leq K \leq 1000$).

- Dòng 2: Ghi N số nguyên a_i . Các số được ghi cách nhau ít nhất một dấu cách ($0 < i < N$, $0 < a_i < 10^4$).

Dữ liệu ra: Ghi ra file văn bản SIGN.OUT theo cấu trúc như sau:

- Dòng 1: Ghi số 0 nếu không có cách đặt dấu phù hợp, ngược lại ghi số 1 tiếp đó ghi liền (không dùng dấu cách) một cách đặt dấu phù hợp tương ứng với các số hạng trong dãy chia hết cho K .

Ví dụ:

SIGN.INP	SIGN.OUT
4 7	1-++
1 2 3 5	

SIGN.INP	SIGN.OUT
4 6	0
1 2 3 5	

(Chú ý: Các chương trình thực hiện không quá 1 giây đối với tất cả các bộ dữ liệu vào)

==HẾT==