



Ngôn ngữ lập trình C++

\$01\$ BÀI TẬP LẬP TRÌNH NÂNG CAO



1. Lại bài toán cái túi

Có N đồ vật, và cái túi có sức chứa không quá M đơn vị khối lượng. Đồ vật thứ i có khối lượng là A_i và giá trị là C_i với $i = 1, 2, \dots, N$.

Yêu cầu: Hãy sắp xếp các đồ vật vào các túi sao cho tổng khối lượng các vật bỏ vào không quá M và tổng giá trị của các vật đó lớn nhất có thể.

Dữ liệu cho trong file **KNAPAGAIN.INP** gồm:

- Dòng đầu ghi hai số nguyên dương N và M tương ứng là số đồ vật và sức chứa tối đa của cái túi.
- Dòng thứ hai ghi N số nguyên dương A_1, A_2, \dots, A_N là khối lượng của N đồ vật.
- Dòng thứ ba ghi N số nguyên dương C_1, C_2, \dots, C_N là giá trị của N đồ vật.

Kết quả đưa ra file **KNAPAGAIN.OUT** là tổng giá trị lớn nhất có thể của các vật được bỏ vào túi.

Ví dụ:

KNAPAGAIN.INP	KNAPAGAIN.OUT
5 10	20
1 2 2 9 10	
1 1 2 3 20	

Giới hạn:

- Sub 1: $N \leq 20$; $A_i, C_i \leq 10^8$; $M \leq 10^9$;
- Sub 2: $N \leq 40$; $A_i, C_i \leq 10^8$; $M \leq 10^9$;
- Sub 3: $N \leq 1000$; $A_i, C_i \leq 10^4$; $M \leq 10^4$;
- Sub 4: $N \leq 1000$; $A_i \leq 10^8$; $C_i \leq 50$; $M \leq 10^9$;



2. Đường đi trên lưới

Cho lưới ô vuông kích thước $N \times N$. Ô trên trái có tọa độ $(1, 1)$, ô dưới phải có tọa độ (N, N) . Trên mỗi ô có ghi một số nguyên. Bạn phải đi từ ô trên trái xuống ô dưới phải theo các quy tắc sau:

- Không được ra ngoài lưới,
- Chỉ được đi sang phải hoặc xuống dưới theo các ô kề cạnh,
- Mỗi ô đi qua không quá một lần,
- Không được đi qua quá K ô chứa giá trị âm.

Yêu cầu: Cho N, K ($0 < N \leq 1000$; $0 \leq K \leq 50$) và giá trị các ô trên lưới (số nguyên thuộc kiểu Integer). Hãy xác định tổng lớn nhất của đường đi theo quy tắc trên (tổng đường đi là tổng các ô trên đường đi). Nếu không có đường đi thỏa mãn thì đưa ra thông báo **NO**.

Dữ liệu: Vào từ file văn bản GRID.INP gồm:

- Dòng đầu tiên chứa 2 số nguyên N, K .
- N dòng sau: mỗi dòng chứa N số nguyên mô tả một dòng của lưới, lần lượt từ trên xuống dưới.

Kết quả: Đưa ra file văn bản GRID.OUT, là kết quả tìm được.

Ví dụ:

GRID.INP				
4	1			
1	2	3	-5	
-10	6	0	-1	
-10	-10	-10	2	
0	0	0	1	

GRID.OUT	
11	

1	2	3	-5
-10	6	0	-1
-10	-10	-10	2
0	0	0	1

Giới hạn: Có 50% số test ứng với 50% số điểm của bài là $n \leq 100$; $k \leq 10$.

**3. Dãy con không giảm dài nhất khi đổi dấu không quá 3 số hạng**

Cho dãy số nguyên a_1, a_2, \dots, a_n . Bằng cách đổi dấu không quá 3 số hạng, hãy tìm dãy con tăng dài nhất của dãy đã cho, tức là tìm dãy chỉ số i_1, i_2, \dots, i_k sao cho:

- $1 \leq i_1 < i_2 < \dots < i_k \leq n$;
- Dãy con $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ là dãy không giảm nếu đổi dấu không quá 3 số hạng của dãy này;
- k có giá trị lớn nhất có thể.

Dữ liệu cho trong file **SEQD.INP** như sau:

- Dòng thứ nhất ghi số nguyên dương n ($n \leq 1000$);
- Dòng thứ hai ghi n số nguyên a_1, a_2, \dots, a_n ($|a_i| \leq 10^9$).

Kết quả ghi ra file **SEQD.OUT** là độ dài của xâu con dài nhất tìm được.

Ví dụ:

SEQD.INP	SEQD.OUT	SEQD.INP	SEQD.OUT
5 9 1 2 3 4	5	6 9 9 1 0 -22 33	5

**4. Bài toán đồng sỏi**

Có n đồng sỏi xếp thành một hàng, đồng thứ i có A_i viên sỏi. Ta có thể ghép hai đồng sỏi kề nhau thành một đồng và mất một chi phí bằng tổng hai đồng sỏi đó. Hãy tìm cách ghép n đồng sỏi này thành một đồng với chi phí là nhỏ nhất.

Ví dụ: Với 5 đồng sỏi:

$$\begin{array}{r}
 4 \quad \underline{1 \quad 2} \quad 7 \quad 5 \\
 \underline{4 \quad 3} \quad 7 \quad 5 \\
 7 \quad \quad \underline{7 \quad 5} \\
 \underline{7 \quad \quad 12} \\
 19 \qquad \leftarrow \text{đồng cuối cùng}
 \end{array}$$

Chi phí nhỏ nhất: $3 + 7 + 12 + 19 = 41$.

Dữ liệu vào cho trong file **Donsoi.inp** có dạng:

- Dòng đầu tiên là số nguyên n ($n \leq 500$) là số đồng sỏi.
- Dòng thứ hai là n số nguyên là số sỏi của n đồng. ($0 < A_i \leq 100000$).

Dữ liệu ra ghi trong file **Donsoi.out** là một số nguyên duy nhất là chi phí nhỏ nhất.

Ví dụ:

Donsoi.Inp	Donsoi.Out
5 4 1 2 7 5	41



5. Cắt lưới

Cho lưới ô vuông gồm N dòng và M cột. Hãy cắt lưới ô vuông này theo các đường chia của lưới sao cho các phần tách ra sau khi cắt là các hình vuông. Chú ý là khi cắt theo các đường chia của lưới thì cắt hết đường chia đó.

Yêu cầu: Tìm cách cắt với số lần cắt là ít nhất.

Dữ liệu cho trong file **CutNet.Inp** gồm hai số nguyên dương N và M ($N, M \leq 500$).

Kết quả ghi trong file **CutNet.Out** là số lần cắt ít nhất cần thực hiện.

Ví dụ:

CutNet.Inp	CutNet.Out
1 4	3