



Ngôn ngữ lập trình C++

BÀI TẬP NÂNG CAO



1. Truy vấn tổng liên tiếp – QuerySC.Cpp

Cho hai số nguyên a, b ($a \leq b$) và số nguyên dương k . Ta xét các số nguyên không lớn hơn b và có dạng: $a, a + k, a + 2k, \dots, a + t.k \dots$ Hãy tính tổng các số hạng này. Giá trị tổng này ta đặt $S(a, b)$.

Dữ liệu cho trong file **QuerySC.Inp** gồm:

- Dòng đầu ghi số nguyên dương Q ($Q \leq 2 \times 10^5$) là số lần yêu cầu tính tổng.
- Q dòng tiếp theo, mỗi dòng ghi cặp số nguyên a, b và k ($-10^8 \leq a \leq b \leq 10^8, 1 \leq k$).

Kết quả ghi ra file **QuerySC.Out** gồm Q dòng, mỗi dòng ghi tổng $S(a, b)$ tương ứng.

Ví dụ:

QuerySC.INP	QuerySC.OUT	Giải thích
2	9	$a = 1, b = 6, k = 2$:
1 6 2	0	Các số hạng: 1, 3, 5. Tổng bằng 9.
-2 2 1		$a = -2, b = 2, k = 1$:
		Các số hạng: -2, -1, 1, 0, 1, 2. Tổng bằng 0.



2. Tổng cặp – SumPairs.Cpp

Cho dãy số nguyên A_1, A_2, \dots, A_N . Ta xây dựng dãy B_1, B_2, \dots, B_M với $M = N*(N-1)/2$; các số hạng trong dãy B có dạng $A_i + A_j$ với $1 \leq i < j \leq N$.

Yêu cầu: Đưa ra dãy B sau khi đã sắp xếp tăng dần.

Dữ liệu cho trong file **SumPairs.Inp** gồm:

- Dòng đầu ghi số nguyên dương N ($N \leq 1000$).
- Dòng thứ 2 ghi N số nguyên A_1, A_2, \dots, A_N ($|A_i| \leq 10^9$).

Kết quả ghi ra file **SumPairs.Out** gồm một dòng là các số hạng của dãy B đã được sắp xếp tăng dần.

Ví dụ:

SumPairs.Inp	SumPairs.Out
4	4 4 4 6 6 6
1 3 3 3	

**3☀. Tổng các số liên tiếp liền trước lớn nhất (1) – SumCMax1.Cpp**

Cho dãy gồm có N số nguyên A_1, A_2, \dots, A_N và chỉ số k . Hãy tìm các số hạng kề liên tiếp và đứng trước A_k để tổng có giá trị lớn nhất. Tức là tìm chỉ số i ($1 \leq i \leq k$) để tổng $A_i + A_{i+1} + \dots + A_k$ có giá trị lớn nhất.

Dữ liệu cho trong file **SumCMax1.Inp** gồm:

- Dòng thứ nhất ghi hai số nguyên dương N là số các số hạng của dãy và chỉ số k .
- Dòng thứ hai ghi N số nguyên A_1, A_2, \dots, A_N .

Kết quả ghi ra file **SumCMax1.Out** là giá trị lớn nhất của tổng có thể nhận được.

Ví dụ:

SumCMax1.Inp	SumCMax1.Out	Giải thích
4 3 -10 2 2 10	4	$k = 3$. Ta chọn $A[2]$ và $A[3]$ tổng $2 + 2 = 4$.

Giới hạn:

- $1 \leq k \leq N \leq 10^5$;
- $|A_i| \leq 10^9$.

**4☀. Tổng các số liên tiếp liền trước lớn nhất (2) – SumCMax2.Cpp**

Cho dãy gồm có N số nguyên A_1, A_2, \dots, A_N . Có Q yêu cầu, mỗi yêu cầu cho chỉ số k . Hãy tìm các số hạng kề liên tiếp và đứng trước A_k để tổng có giá trị lớn nhất. Tức là tìm chỉ số i ($1 \leq i \leq k$) để tổng $A_i + A_{i+1} + \dots + A_k$ có giá trị lớn nhất. Giá trị lớn nhất này ta đặt $S(k)$.

Dữ liệu cho trong file **SumCMax2.Inp** gồm:

- Dòng thứ nhất ghi hai số nguyên dương N là số các số hạng của dãy.
- Dòng thứ hai ghi N số nguyên A_1, A_2, \dots, A_N .
- Dòng thứ ba ghi số nguyên Q là số lần yêu cầu.
- Dòng thứ tư ghi Q chỉ số k_1, k_2, \dots, k_Q .

Kết quả ghi ra file **SumCMax2.Out** gồm Q dòng. Dòng thứ i ghi giá trị $S(k_i)$.

Ví dụ:

SumCMax2.Inp	SumCMax2.Out
4	2
-10 2 2 10	4
2	
2 3	

Giới hạn:

- $|A_i| \leq 10^9$;
- Có 50 % số test ứng với $1 \leq Q \leq N \leq 10^3$;
- Có 50 % số test ứng với $1 \leq Q \leq N \leq 2 \times 10^5$;



5. Bỏ sỏi vào, lấy sỏi ra – PPStone.Cpp

Có ba chiếc hộp lớn, ta gọi là hộp A , hộp B và hộp C . Ban đầu, hộp A có N hòn sỏi, hộp B và hộp C không có viên sỏi nào. Các viên sỏi được đánh số từ 1 đến N , viên sỏi thứ i có khối lượng A_i (microgram).

Một con robot được lập trình để lấy các viên sỏi bằng hai loại lệnh:

Loại 1 có dạng 2 số nguyên: 1 và i . Khi thực hiện lệnh này, robot sẽ lấy viên sỏi thứ i trong hộp A và bỏ vào hộp B .

Loại 2 có dạng 1 số nguyên: 2. Khi thực hiện lệnh này, robot sẽ lấy một viên sỏi trong hộp B và có khối lượng lớn nhất. Viên sỏi được lấy ra sẽ bỏ vào hộp C .

Yêu cầu: Cho biết danh sách các lệnh được thực hiện bởi robot (các lệnh được thực hiện tuần tự theo danh sách), hãy đưa ra danh sách khối lượng các viên sỏi **lần lượt** được bỏ vào hộp C .

Dữ liệu cho trong file **PPStone.Inp** gồm:

- Dòng đầu ghi số nguyên dương N là số viên sỏi ban đầu trong hộp A .
- Dòng thứ hai ghi N số nguyên dương A_1, A_2, \dots, A_N ($A_i \leq 10^8$) là khối lượng của N viên sỏi.
- Dòng thứ ba ghi số nguyên dương Q là số lệnh mà robot sẽ thực hiện.
- Q dòng cuối, mỗi dòng ghi một lệnh thuộc tập 2 loại lệnh trên.

Kết quả ghi ra file **PPStone.Out** là danh sách khối lượng các viên sỏi **lần lượt** được bỏ vào hộp C .

Ví dụ:

PPStone.Inp	PPStone.Out	Giải thích
5	2	Lệnh 1: Hộp $B = \{2\}$, hộp $C = \{\}$.
1 9 2 8 100	9	Lệnh 2: Hộp $B = \{2, 1\}$, hộp $C = \{\}$.
5		Lệnh 3: Hộp $B = \{1\}$, hộp $C = \{2\}$.
1 3		Lệnh 4: Hộp $B = \{1, 9\}$, hộp $C = \{2\}$.
1 1		Lệnh 5: Hộp $B = \{1\}$, hộp $C = \{2, 9\}$.
2		
1 2		
2		

Giới hạn:

- Có 50% số test ứng với $Q, N \leq 1000$;
- Có 50% số test ứng với $Q, N \leq 200.000$.

**6. Máy cắt gỗ - CwoodM.Cpp**

Có một thanh gỗ dài X (mm). Bạn cần sử dụng cái máy cắt gỗ để cắt thanh gỗ thành N đoạn có độ dài là d_1, d_2, \dots, d_N . Chú ý là $d_1 + d_2 + \dots + d_N = X$.

Rõ ràng là cần thực hiện cắt $N - 1$ lần. Khi thực hiện cắt một lần nào đó, ta sẽ lấy 1 đoạn và cắt chúng thành 2 đoạn có độ dài nhỏ hơn.

Vấn đề ở đây là, khi chọn một đoạn có độ dài a , để cắt thành hai đoạn thì chi phí cho lần cắt đó là a .

Yêu cầu: Tìm cách cắt để tổng chi phí thực hiện là nhỏ nhất.

Dữ liệu cho trong file **CwoodM.Inp** gồm:

- Dòng đầu ghi 2 số nguyên dương X và N .
- Dòng thứ 2 ghi N số nguyên dương d_1, d_2, \dots, d_N . Dữ liệu đảm bảo $d_1 + d_2 + \dots + d_N = X$.

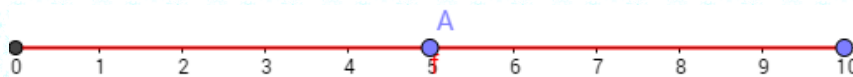
Kết quả ghi ra file **CwoodM.Out** là tổng chi phí nhỏ nhất có thể đạt được.

Ví dụ:

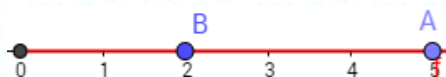
CwoodM.Inp	CwoodM.Out	Giải thích
10 3 3 5 2	15	

Giải thích:

- Lần 1 cắt ở vị trí A, mất chi phí 10.



- Lần 2 cắt ở vị trí B, mất chi phí 5.



Tổng chi phí bằng $10 + 5 = 15$.

Giới hạn:

- Có 40% số test ứng với $N \leq 4$;
- Có 60% số test ứng với $N \leq 200000$;
- $N \leq X \leq 10^8$.



7. Truy vấn tổng lũy thừa – QueryPower.Cpp

Cho hai số nguyên a và b ($0 \leq a \leq b \leq 10^6$). Ta gọi $S(a, b) = a^{2021} + (a+1)^{2021} + \dots + b^{2021}$.

Yêu cầu: Hãy tính $S(a, b) \% (10^9 + 7)$.

Dữ liệu cho trong file **QueryPower.Inp** gồm:

- Dòng đầu ghi số nguyên dương N là số lần tính tổng ($N \leq 200000$).
- N dòng sau, mỗi dòng ghi 2 số nguyên a và b .

Kết quả ghi ra file **QueryPower.Out** gồm N dòng. Mỗi dòng ghi giá trị $S(a, b) \% (10^9 + 7)$ tương ứng.

Ví dụ:

QueryPower.Inp	QueryPower.Out
2	1
0 1	1
1 1	

Giới hạn:

- Có 50% số test ứng với $N \leq 1000$; $0 \leq a \leq b \leq 10000$;