



Ngôn ngữ lập trình C++

BÀI TẬP LẬP TRÌNH CƠ BẢN VÀ NÂNG CAO

A. Phần cơ bản



1. Số bằng K – EquaK1.Cpp

Cho dãy số nguyên A_1, A_2, \dots, A_N và số K . Đếm xem trong dãy có bao nhiêu số hạng bằng K .

Dữ liệu cho trong file **EquaK1.Inp** gồm:

- Dòng 1 ghi số nguyên dương N và K .
- Dòng 2 ghi số nguyên A_1, A_2, \dots, A_N .

Kết quả ghi ra file **EquaK1.Out** là số các số trong dãy bằng K .

Ví dụ:

EquaK1.Inp	EquaK1.Out
4 9 9 4 4 9	2

Giới hạn:

- $N \leq 10^5$;
- $|K|, |A_i| \leq 10^6$.



2. Cặp số có tổng bằng K - PairEquaK1.Cpp

Cho dãy số nguyên A_1, A_2, \dots, A_N và số K . Đếm xem trong dãy có bao nhiêu cặp số hạng a_i và a_j sao cho $a_i + a_j = K$ với $(1 \leq i < j \leq N)$.

Dữ liệu cho trong file **PairEquaK1.Inp** gồm:

- Dòng 1 ghi số nguyên dương N và K .
- Dòng 2 ghi số nguyên A_1, A_2, \dots, A_N .

Kết quả ghi ra file **PairEquaK1.Out** là số các số trong dãy bằng K .

Ví dụ:

EquaK1.Inp	EquaK1.Out	Giải thích
4 13 9 4 3 10	2	$A[1] + A[2] = 9 + 4 = 13$ $A[3] + A[4] = 3 + 10 = 13$

Giới hạn:

- $N \leq 10^3$;
- $|K|, |A_i| \leq 10^6$.



3. Lấy táo (1)

Có N quả táo được xếp thành một hàng ngang. Các quả táo được đánh số thứ tự từ 1 đến N (hướng từ trái sang phải). Quả táo thứ i có khối lượng a_i .

Yêu cầu: Bạn hãy chọn các quả táo ở các vị trí *liên tiếp* nhau sao cho:

- Các quả táo có khối lượng bằng nhau.
- Số lượng các quả táo là nhiều nhất.



Dữ liệu cho trong file **SelectApple1.Inp** gồm:

- Dòng 1 ghi số nguyên dương N là số lượng quả táo ($N \leq 10^5$).
- Dòng 2 ghi N số nguyên dương a_1, a_2, \dots, a_N ($a_i \leq 10^8$).

Kết quả ghi ra file **SelectApple1.Out** là số táo nhiều nhất có thể chọn được.

Ví dụ:

SelectApple1.Inp	SelectApple1.Out
5 1 4 4 3 2	2
5 1 2 3 4 1	1



4. Lấy táo (2)

Có N quả táo được xếp thành một hàng ngang. Các quả táo được đánh số thứ tự từ 1 đến N (hướng từ trái sang phải). Quả táo thứ i có khối lượng a_i .

Yêu cầu: Bạn hãy chọn các quả táo sao cho:

- Các quả táo có khối lượng bằng nhau (có thể không ở các vị trí liên tiếp).
- Số lượng các quả táo là nhiều nhất.

Dữ liệu cho trong file **SelectApple2.Inp** gồm:

- Dòng 1 ghi số nguyên dương N là số lượng quả táo ($N \leq 10^5$).
- Dòng 2 ghi N số nguyên dương a_1, a_2, \dots, a_N ($a_i \leq 10^8$).

Kết quả ghi ra file **SelectApple2.Out** là số táo nhiều nhất có thể chọn được.

Ví dụ:

SelectApple2.Inp	SelectApple2.Out
5 1 4 4 3 2	2
5 1 2 3 4 1	2

A. Phần Nâng cao



1. Cắt thanh gỗ

Một thanh gỗ có độ dài L (cm). John muốn cắt thanh gỗ này thành 12 đoạn nhỏ (mỗi đoạn có độ dài nguyên) bằng 11 lần cắt. Tính xem John có bao nhiêu cách cắt như vậy. Hai cách cắt được gọi là khác nhau nếu có một vị trí trên thanh gỗ bị cắt ở cách này nhưng không thuộc cách cắt khác.

Dữ liệu cho trong file **CutWood12.Inp** gồm một số nguyên dương L là độ dài khúc gỗ ($12 \leq L \leq 200$).

Kết quả ghi ra file **CutWood12.Out** là số cách cắt gỗ.

Ví dụ:

CutWood12.Inp	CutWood12.Out
12	1

**2. Không chia hết**

Cho 4 số nguyên dương A, B, C, D . Tính số các số nguyên thuộc đoạn $[A, B]$ không chia hết cho C và cũng không chia hết cho D .

Dữ liệu cho trong file **AntiDivision.Inp** gồm 4 số nguyên dương A, B, C, D .

Kết quả ghi ra file **AntiDivision.Out** là số các số nguyên thuộc đoạn $[A, B]$ không chia hết cho C và cũng không chia hết cho D .

Ví dụ:

AntiDivision.Inp	AntiDivision.Out	Giải thích
4 9 2 3	2	Có 5 số thuộc $[4, 9]$ 4, 5, 6, 7, 8, 9, trong đó có 2 số 5 và 7 đều không chia hết cho 2 và 3.

Giới hạn:

- $1 \leq A \leq B \leq 10^{18}$;
- $1 \leq C \leq D \leq 10^9$.

**3. Xếp công việc**

Có N công việc được đánh số thứ tự từ 1 đến N . Công việc thứ i có thời gian hoàn thành là A_i và hạn cuối cần hoàn thành là B_i . Nếu công việc i hoàn thành sau thời điểm B_i thì ta nói hoàn thành không đúng hạn. Có 3 người cần thực hiện N công việc này, người 1 thực hiện công việc 1 đến công việc X ; người 2 thực hiện công việc $X + 1$ đến công việc Y ; người 3 thực hiện công việc $Y + 1$ đến công việc N . Ba người đều thực hiện công việc của mình một cách độc lập.

Với mỗi người, khi thực hiện các công việc cần tuân thủ đó là: Không được thực hiện một lúc nhiều công việc. Thực hiện công việc này rồi mới chuyển sang thực hiện công việc khác. Vấn đề đặt ra với mỗi người là: Tìm cách sắp xếp các công việc thực hiện sao cho các công việc đều hoàn thành đúng thời hạn. Mỗi người có thể bắt đầu từ thời điểm 0.

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

- Dòng đầu ghi số nguyên dương N là số công việc.
- N dòng sau, dòng thứ i ghi hai số nguyên A_i và B_i
- Dòng cuối ghi 2 số nguyên X, Y .

Kết quả ghi ra file **Work.Out** gồm 3 dòng.

- Dòng 1 ghi “Yes” nếu người 1 có cách xếp thực hiện các công việc đúng hạn tất cả các công việc mình làm. Ngược lại ghi “No”.
- Dòng 2 ghi “Yes” nếu người 2 có cách xếp thực hiện các công việc đúng hạn tất cả các công việc mình làm. Ngược lại ghi “No”.



- Dòng 3 ghi “Yes” nếu người 3 có cách xếp thực hiện các công việc đúng hạn tất cả các công việc mình làm. Ngược lại ghi “No”.

Ví dụ:

Work.Inp	Work.Out	Giải thích
6	Yes	- Người 1 làm công việc 1 và 2: Thứ tự là (2) → (1).
1 3	No	
2 2	Yes	- Người 2 làm công việc 3, 4. Không có cách nào hoàn thành cả hai công việc đúng hạn.
7 8		
6 6		
5 9		- Người 3 làm công việc 5, 6: Thứ tự (5) → (6).
4 4		
2 4		

Giới hạn:

- $3 \leq N \leq 2 \times 10^5$;
- $1 \leq X < Y < N$;
- $1 \leq A_i \leq B_i \leq 10^6$.



4. Tìm số

Cho số tự nhiên X . Tìm số tự nhiên Y sao cho:

- Tất cả các chữ số của Y đều khác nhau.
- Tổng các chữ số của Y bằng X .
- Y có giá trị nhỏ nhất.

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

- Dòng đầu ghi số nguyên dương T ($1 \leq T \leq 10$) là số bộ dữ liệu;
- T dòng tiếp theo, mỗi dòng ghi một số nguyên X ($1 \leq X \leq 50$).

Kết quả ghi ra file **UniqueNumber.Out** gồm T dòng, mỗi dòng ghi số nguyên Y tương ứng tìm được. Nếu không có giá trị Y thỏa mãn thì ghi -1.

Ví dụ:

UniqueNumber.Inp	UniqueNumber.Out
3	2
2	19
10	-1
50	



5. Bước nhảy

Bạn cần điều khiển một con robot dịch chuyển trên một trục tọa độ. Ban đầu, robot ở tại vị trí tọa độ 0 (gốc tọa độ). Bàn phím điều khiển có hai nút lệnh: nút “Lùi” và nút “Tiến”. Nếu hiện tại robot đang ở vị trí tọa độ Y , và đã bấm được k lần (ban đầu $k = 0$).

- Khi bấm vào nút “Lùi” nó sẽ dịch chuyển đến tọa độ $Y - 1$.
- Khi bấm vào nút “Tiến” nó sẽ dịch chuyển đến tọa độ $Y + k + 1$.



Chú ý là, sau mỗi lần bấm, số k sẽ tăng lên 1.

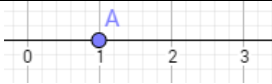
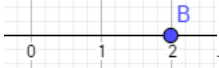
Yêu cầu: Hãy tìm cách bấm với số lần ít nhất để đưa robot đến vị trí có tọa độ là X .

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

- Dòng đầu ghi số nguyên dương T ($T \leq 1000$) là số bộ dữ liệu.
- T dòng tiếp theo, mỗi dòng ghi một số nguyên dương X ($X \leq 10^6$).

Kết quả ghi ra file **JumpX.Out** gồm T dòng, mỗi dòng ghi số lần bấm ít nhất để di chuyển robot đến tọa độ X trong bộ dữ liệu tương ứng.

Ví dụ:

JumpX.Inp	JumpX.Out	Giải thích
2	1	 Bấm 1 lần vào nút “Tiến”.
1	3	
2		 Bấm nút “Tiến”, “Lùi”, “Tiến”.



6. Rót nước

Có N cốc nước được xếp thành một hàng ngang. Các cốc được đánh số thứ tự từ 1 đến N (theo hướng từ trái sang phải). Cốc nước thứ i có lượng nước là A_i (ml). Cần biến đổi dãy cốc nước để lượng nước các cốc bằng nhau theo quy tắc.

Chọn một chỉ số i ($1 \leq i \leq N$), đổ hết lượng nước ở cốc i sang một trong hai cốc kề, sau đó đưa cốc thứ i ra khỏi hàng (mỗi lần thực hiện, số cốc giảm đi 1).

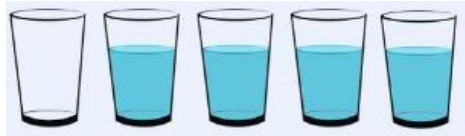
Yêu cầu: Tính số cốc nước nhiều nhất có thể nhận được trong dãy cuối cùng (dãy cốc mà lượng nước trong đó đều bằng nhau).

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

- Dòng đầu ghi số nguyên dương T là số bộ dữ liệu. Mỗi bộ dữ liệu gồm:
- Dòng thứ nhất ghi số nguyên dương N là số cốc nước.
- Dòng thứ hai ghi N số nguyên dương A_1, A_2, \dots, A_N .

Kết quả ghi ra file **MoveWater.Out** gồm T dòng, mỗi dòng ghi số cốc nhiều nhất có thể nhận được khi biến đổi đưa về lượng nước trong tất cả các cốc bằng nhau.

Ví dụ:

MoveWater.Inp	MoveWater.Out	Giải thích
2 5 2 3 5 5 5 3 1 3 5	4 1	<p>- Bộ dữ liệu 1: Lấy cốc có lượng nước là 2 đổ vào cốc có lượng nước là 3. Ta được 4 cốc có lượng nước là 5.</p>  <p>- Bộ dữ liệu 2: Ta đổ các cốc lại để được 1 cốc có lượng nước là 9.</p>

Giới hạn:

- $1 \leq A_i \leq 10^5$;
- $1 \leq T, N \leq 3.000$;
- Tổng giá trị của tất cả N trong mỗi test không quá 3.000.