



Ngôn ngữ lập trình C++

\$06\$ BÀI TẬP LẬP TRÌNH NÂNG CAO



1☀. 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 |



2☀. Quả Táo

Một công ty đa quốc gia đang hỏi bạn để giúp họ sửa đổi di truyền của một giống táo. Để cho táo phát triển nhanh hơn, có được nhiều quả hơn và các quả to và đẹp hơn. Để làm điều đó, cần phải chèn vào DNA của táo một số kí tự thuộc {'A', 'C', 'G', 'T'} để trong DNA của táo chứa một đoạn gen đặc biệt (tức là sau khi chèn, trong DNA có một xâu con gồm các kí tự liên tiếp bằng gen đặc biệt). DNA của táo được biểu diễn bởi một dãy các kí tự thuộc tập {'A', 'C', 'G', 'T'}. Việc chèn các kí tự vào DNA của táo đều mất một kinh phí nhất định.

Yêu cầu: Cho biết DNA của táo, đoạn gen đặc biệt và chi phí chèn mỗi lại kí tự. Hãy tìm cách chèn các kí tự vào DNA để xuất hiện đoạn gen đặc biệt với tổng chi phí là nhỏ nhất.

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

- Dòng đầu tiên ghi dãy kí tự thuộc tập {'A', 'C', 'G', 'T'} biểu diễn DNA của táo (số kí tự không quá 10000).
- Dòng thứ hai ghi dãy kí tự thuộc tập {'A', 'C', 'G', 'T'} biểu diễn đoạn gen đặc biệt (số kí tự không quá 5000).
- Dòng thứ ba ghi bốn số nguyên thuộc $[0; 1000]$ tương ứng là chi phí khi chèn một kí tự 'A', 'C', 'G', 'T' vào DNA của táo.

Kết quả ghi ra file **APPLE.OUT** gồm một số là tổng chi phí nhất tìm được.

Ví dụ:



| APPLE.INP | APPLE.OUT | APPLE.INP | APPLE.OUT | APPLE.INP | APPLE.OUT |
|-----------|-----------|-----------|-----------|-------------|-----------|
| GTA | 10 | TATA | 3 | TCGCGAG | 25 |
| CAT | | CACA | | TGCAG | |
| 5 7 1 3 | | 3 0 3 0 | | 10 10 15 15 | |

**3☀️. Trò chơi bốc bài**

Xét trò chơi giữa hai người: Trên bàn có n lá bài xếp thành một hàng, trên mỗi lá bài có ghi một số nguyên dương Q_i (các lá bài được đánh số từ 1 đến n theo hướng từ trái qua phải). Hai người luân phiên nhau chơi, mỗi người đến lượt mình chọn một trong hai quân bài ngoài cùng (trái nhất hoặc phải nhất) và bốc quân bài đó lên. Trò chơi kết thúc khi không còn quân bài nào trên bàn và người thắng cuộc là người có tổng các số ghi trên các quân bài mình bốc lên lớn hơn (nếu hai người có tổng quân bài bằng nhau thì hòa).

Yêu cầu: Cho biết người đi đầu có chắc thắng không?

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

- Dòng đầu ghi số nguyên dương T là số testcase ($T \leq 10$);
- Với mỗi testcase gồm hai dòng;
- Dòng thứ nhất ghi số nguyên dương n ($n \leq 1000$);
- Dòng thứ hai ghi n số nguyên dương ($Q_i \leq 10^9$).

Kết quả ghi trong file **GAME.OUT** gồm T dòng, mỗi dòng là **YES** nếu người đi trước tương ứng với testcase đó là chắc thắng, ngược lại ghi **NO**.

Ví dụ:

| GAME.INP | GAME.OUT |
|----------|----------|
| 2 3 | NO |
| 1 5 3 | YES |
| 4 | |
| 1 6 2 3 | |

**4☀️. Phần thưởng (VOI2020)**

Alice vừa đoạt giải quán quân trong một kì thi lập trình danh giá. Ban tổ chức trao thưởng thông qua một trò chơi như sau: Có n thẻ xếp trên một hàng dài, trên mỗi thẻ viết một số nguyên dương. Ban tổ chức cho phép Alice thực hiện nhiều bước để chọn đúng ra k cặp thẻ, mỗi bước thực hiện theo một trong các quy tắc sau:

1. Chọn 2 thẻ đầu hàng;
2. Chọn 2 thẻ cuối hàng;
3. Chọn 1 thẻ đầu hàng và 1 thẻ cuối hàng;
4. Loại 1 thẻ đầu hàng ra khỏi hàng;
5. Loại 1 thẻ cuối hàng ra khỏi hàng.

Sau mỗi bước, nếu chọn được hai thẻ thì loại 2 thẻ đó ra khỏi hàng và Alice nhận được số tiền thưởng bằng giá trị tuyệt đối của hiệu hai số ghi trên hai thẻ đó.



Yêu cầu: Hãy giúp Alice tìm cách chọn đúng k cặp thẻ để đạt được tổng số tiền thưởng là lớn nhất.

Dữ liệu: vào từ file BONUS.INP gồm:

- Dòng thứ nhất chứa hai số nguyên dương n và k ($2 \times k \leq n$).
- Dòng thứ hai ghi n số nguyên dương là giá trị ghi trên từng thẻ, mỗi thẻ một số tương ứng lần lượt từ đầu hàng. Các số có giá trị không quá 10^9 .

Kết quả: Ghi ra file BONUS.OUT một số nguyên duy nhất là tổng số tiền thưởng lớn nhất tìm được.

Ràng buộc:

- Có 40% số test ứng với $n \leq 300; k \leq 2$;
- Có 40% số test khác ứng với $n \leq 30, 2 \times k = n$;
- 20% số test còn lại ứng với $n \leq 300$.

Ví dụ:

| BONUS.INP | BONUS.OUT |
|---------------------|-----------|
| 6 2 1 3 10 2 1 4 | 12 |

Giải thích:

- Bước 1: Chọn hai thẻ cuối hàng là 1 và 4. Nhận được $|4 - 1| = 3$.
- Bước 2: Loại thẻ cuối hàng có giá trị 2;
- Bước 3: Chọn thẻ đầu hàng và cuối hàng có giá trị 1 và 10. Nhận được $|10 - 1| = 9$.

Tổng số tiền nhận được: $3 + 9 = 12$.



5 Chia kẹo

Có N em bé được đánh chỉ số $1, 2, \dots, N$. Có K chiếc kẹo sẽ chia cho em bé. Em bé thứ i sẽ nhận được số kẹo thuộc: $0, 1, 2, \dots, a_i$. Chú ý rằng tổng số kẹo nhận được của N em bé luôn bằng K .

Yêu cầu: Tính số cách chia kẹo thỏa mãn trên.

Dữ liệu cho trong file DivideCand.Inp gồm:

- Dòng đầu ghi hai số nguyên dương N và K .
- Dòng thứ hai ghi N số nguyên a_1, a_2, \dots, a_N .

Kết quả ghi ra file DivideCand.Out là số dư khi chia số cách chia thỏa mãn yêu cầu cho 10^9+7 .

Ví dụ:

| DivideCand.Inp | DivideCand.Out | Giải thích |
|----------------|----------------|---|
| 3 4 1 2 3 | 5 | 0, 1, 3 0, 2, 2 1, 0, 3 1, 1, 2 1, 2, 1 |

Giới hạn:

- $1 \leq N \leq 100$;
- $0 \leq a_i \leq K \leq 10^5$.