

1 Lý thuyết

1.1 Độ phức tạp thuật toán

$n \geq 1$

$$\begin{aligned}1 + 2 + 3 + \dots + n &= \frac{n(1+n)}{2} \\1 + 2^2 + 3^2 + \dots + n^2 &= \frac{n(n+1)(2n+1)}{6} \\1 + 2^3 + 3^3 + \dots + n^3 &= \frac{n^2(n+1)^2}{4} \\1 + a + a^2 + \dots + a^n &= \frac{a^{n+1} - 1}{a - 1}\end{aligned}$$

$n \leq 2$

$$u_n = u_1 + (n-1)d$$

2 Bài tập

2.1 Cấu trúc dữ liệu

1. Bạn được giáo viên giao cho công việc quản lý điểm thực hành của cả lớp (40 sinh viên) bao gồm điểm thi dạng số và mã số sinh viên, ví dụ: 10 và 616263.
Ý tưởng của bạn là tạo ra 2 mảng: "int diemso[40]" và "int masosv[40]" để lưu trữ. Tuy nhiên phương pháp này rất tốn kém (thời gian, bộ nhớ) khi bạn muốn thêm, xóa, sửa điểm số của sinh viên.
Nhiệm vụ của bạn là tạo ra 1 kiểu dữ liệu mới để lưu trữ điểm thực hành theo yêu cầu của giáo viên.
2. 1 team liên quân gồm có 6 người (5 chính, 1 dự bị), mỗi người có 1 sở trường riêng như đi rừng, mid, top,... và mỗi người chỉ có thể dụng tốt 1 tướng. Nhiệm vụ của bạn là tạo ra một kiểu dữ liệu để lưu trữ thông tin các thành viên trong team của bạn.
3. Bạn là một chuyên gia kinh doanh bất động sản, bạn có thể định giá và dự đoán giá của 1 mảnh đất với độ chính xác rất cao. Tuy nhiên bạn đang dự đoán và định giá rất nhiều mảnh đất khắp nơi trên thế giới, vì vậy bạn cần phải lưu trữ thông tin của tất cả mảnh đất này.

Hãy tạo ra một kiểu dữ liệu để lưu trữ thông tin này (Ít nhất 3 thông tin). Nhập và in ra thông tin cho 2 mảnh đất mà bạn ưng ý nhất.

4. Tiếp theo bài (3), bạn vẫn là một chuyên gia kinh doanh bất động sản, nhưng vì Covid-19 nên mọi dự đoán của bạn đã sai hoàn toàn, bây giờ bạn chỉ còn lại dưới 10 mảnh đất. Lúc này giá đất của bạn bị phụ thuộc vào giá vàng thế giới. Hãy thêm 1 thuộc tính vào dữ liệu bạn đã tạo ở bài (3) để thể hiện sự phụ thuộc này.
 - Phòng: Tên phòng, Mã phòng.
 - Nhân sự: Tên nhân sự, mã nhân viên, giới tính, chức vụ.
6. Bạn là HLV của câu lạc bộ bóng đá của trường Đại học Nha Trang gồm 10 cầu thủ. Số áo của mỗi cầu thủ là tùy thích từ 1 đến 99 và bạn chọn ra 1 cầu thủ có số áo tùy thích làm đội trưởng. Hãy tạo một kiểu dữ liệu để lưu trữ số áo và đội trưởng của đội bóng này.

2.2 Danh sách liên kết

1. Danh sách liên kết (Linked list)
 - Khai báo kiểu dữ liệu
 - Tạo mới 1 Node
 - Thêm 1 Node vào đầu, vào cuối, vào vị trí bất kỳ
 - Duyệt danh sách, Tìm kiếm
 - Xóa Node ở đầu, ở cuối, ở vị trí bất kỳ
 - Lấy giá trị ở vị trí bất kỳ
 - Đếm số chẵn, đếm số lẻ, tính tổng,...
2. Ngăn xếp (stack)
 - Thêm (Push)
 - Xóa phần tử ở đỉnh (Pop)
 - Lấy giá trị phần tử ở đỉnh (Top)
 - Kiểm tra rỗng (IsEmpty)

- Kiểm tra đầy (IsFull)
 - Chiều dài stack (Size)
3. Hàng đợi (queue)
- Thêm vào cuối
 - Xóa phần tử ở đầu
 - Kiểm tra rỗng
 - Lấy giá trị ở đầu
4. Dùng stack, đổi số N sang nhị phân
5. Dùng stack, viết chương trình tính giá trị biểu thức hậu tố đọc từ file.
6. Dùng stack, viết chương trình phân tích 1 số thành thừa số nguyên tố theo thứ tự lớn trước nhỏ sau.
Ví dụ: $n = 3960$, hiển thị: $3960 = 11 * 5 * 3 * 3 * 2 * 2 * 2$
7. Bạn là lớp trưởng của một lớp học và bạn được giao nhiệm vụ lập danh sách lớp để giao cho giáo viên của bạn. Danh sách lớp cần có tên và giới tính của sinh viên (nam/nữ).
- Hãy tạo danh sách lớp có X sinh viên.
 - In ra số bạn nam/nữ có trong danh sách.
 - Thêm 1 sinh viên vào giữa danh sách nếu số lớp chẵn và thêm vào cuối nếu số lớp là lẻ.
8. Viết chương trình tạo danh sách gồm N số nguyên tố liên tiếp.
9. Viết chương trình chèn 1 node vào 1 vị trí bất kỳ của danh sách các số nguyên tố mà không làm thay đổi thứ tự của danh sách.
10. Bạn được giao nhiệm vụ quản lý một tủ sách gồm N quyển sách có số trang khác nhau. Bạn cần lưu trữ các thông tin của 1 quyển sách bao gồm: tên sách, năm xuất bản. Nhiệm vụ của bạn là tạo ra danh sách các quyển sách bạn đang quản lý theo thứ tự năm xuất bản giảm dần.
- Input:**
Dòng 1: Số nguyên N quyển sách.
Dòng 2: N số nguyên là năm xuất bản của từng quyển sách.
- Output:**
Danh sách N quyển sách đã sắp xếp.

11. DAYNGO - Dây ngoặc đúng

Có thể định nghĩa khái niệm dây ngoặc đúng dưới dạng đệ quy như sau:

1. $()$ là dây ngoặc đúng
2. C là dây ngoặc đúng nếu $C = (A)$ hay $C = AB$ với A, B là các dây ngoặc đúng.

Ví dụ dây ngoặc đúng: $()$, $(())$, $()()$, $(())()$

Ví dụ dây ngoặc sai: $)()$, $(((((, ()((,)))$, $)()()$

Bạn hãy viết chương trình liệt kê tất cả các dây ngoặc đúng có chiều dài n (n chẵn)

Dữ liệu nhập:

- Là số nguyên n (n chẵn, $2 \leq n \leq 20$)

Dữ liệu xuất: với m là số lượng các dây ngoặc đúng có chiều dài n

- Trong m dòng đầu tiên, mỗi dòng liệt kê một dây ngoặc đúng chiều dài n . Các dây được liệt kê theo thứ tự từ điển: $'(<')'$.

- Dòng cuối cùng: là số m

<http://ntucoder.net/Problem/Details/139>

12. Danh sách

Bạn là một người rất nổi tiếng, danh sách bạn bè trên facebook của bạn rất lớn. Tuy nhiên vì có quá nhiều bạn bè ít tương tác nên bạn quyết định xóa 1 số bạn bè. Bạn sử dụng thuật toán như sau:

Algorithm Delete(Friend):

```
DeleteFriend=false
for i = 1 to Friend.length-1
    if (Friend[i].popularity < Friend[i+1].popularity)
        delete i th friend
        DeleteFriend=true
        break
if(DeleteFriend == false)
    delete the last friend
```

Input:

Dòng đầu tiên là N số bạn bè hiện có và K số bạn sẽ xóa. Dòng thứ 2 là độ tương tác n của N số bạn hiện có.

Output:

In ra độ tương tác của $N - K$ bạn còn lại sau khi xóa.

Constraints:

$1 \leq T \leq 1000$

$1 \leq N \leq 10000$

$$1 \leq K \leq N$$

$$1 \leq n \leq 100$$

Ví dụ:

Input:

3 1

3 100 1

Output:

100 1

Input:

5 2

19 12 3 4 17

Output:

19 12 17

13. Bạn được cung cấp một danh sách liên kết chứa N số nguyên. Bạn đã thực hiện các thao tác ngược trên danh sách như sau:

- Lấy ra tất cả tập con chỉ chứa các số nguyên chẵn liên tiếp.
Ví dụ: nếu danh sách ban đầu là $\{1, 2, 8, 9, 12, 16\}$ thì tập con được lấy là $\{2, 8\}, \{12, 16\}$.
- Đảo ngược các tập con đã lấy $\{8, 2\}, \{16, 12\}$

Nhiệm vụ của các bạn là lấy lại danh sách ban đầu.

Input:

Dòng đầu là số N . Dòng thứ 2 là giá trị của N phần tử của danh sách đã đảo ngược

Output

In ra N phần tử của dãy ban đầu.

Ví dụ:

Input:

9

2 18 24 3 5 7 9 6 12

Output:

24 18 2 3 5 7 9 12 6

14. Cho một chuỗi nhị phân, bạn phải thực hiện một số thao tác trên chuỗi này, trong một thao tác, hãy chọn một chuỗi con có độ dài chẵn không rỗng chỉ chứa 0 hoặc chỉ 1 và xóa nó khỏi chuỗi.
Mục tiêu của bạn là giảm thiểu độ dài cuối cùng của chuỗi sau khi thực hiện một số thao tác. Có thể chuỗi cuối cùng có thể trở nên trống, trong trường hợp đó, hãy in ra "TRONG".

Và có thể chứng minh rằng luôn tồn tại một chuỗi duy nhất với độ dài tối thiểu sau khi thực hiện các phép toán.

Input

Nhập vào chuỗi nhị phân S .

Output

In ra chuỗi theo yêu cầu.

Constraints $1 \leq T \leq 10$

$1 \leq \|S\| \leq 10^5$

Ví dụ:

Input:

101001

Output:

10

Input:

1001

Output:

TRONG

Explanation Trong ví dụ 1, đầu tiên xóa chuỗi con "00", chuỗi trở thành "1011", tiếp theo xóa "11", kết quả thu được là "10".

15. Nhiệm vụ của bạn là xây dựng 1 tháp trong N ngày với các điều kiện sau:

- Mỗi ngày bạn được cấp 1 đĩa với kích thước khác nhau
- Đĩa có kích thước lớn hơn sẽ đặt ở cuối tháp
- Đĩa có kích thước nhỏ hơn sẽ đặt ở đỉnh tháp

Thứ tự trong tháp phải theo cấu trúc như sau: bạn không được đặt đĩa mới ở trên đỉnh của tháp cho đến khi tất cả đĩa lớn hơn đã được đặt đúng vị trí.

Print N dòng biểu thị kích thước của đĩa thứ i có thể đặt trên tháp trong ngày thứ i .

Input

- Dòng 1: N là tổng số đĩa sẽ cho bạn trong N ngày tiếp theo
- Dòng 2: N số nguyên n là kích thước của từng đĩa cấp cho bạn trong ngày thứ i .

Output In ra N dòng. Mỗi dòng thứ i in ra kích thước của đĩa có thể đặt trên đỉnh của tháp theo thứ tự giảm dần kích thước của đĩa. Nếu ngày thứ i không có đĩa nào có thể đặt thì in ra dòng trống.

Constraints

$$1 \leq N \leq 10^6$$

$$1 \leq n \leq N$$

Ví dụ:

Input:

5

4 5 1 2 3

Output:

5 4

3 2 1

16. Bạn đang làm việc trong 1 công ty lập trình có quy định kỳ lạ là 1 ngày làm việc H giờ và có thể có nhiều hoặc chỉ có 1 nhân viên làm việc trong ngày, thời gian đến công ty làm việc là tùy ý. Mỗi ngày công ty chọn ra 1 lập trình viên để làm leader trong khoảng thời gian từ S đến E giờ và leader sẽ chấp nhận Q lập trình viên làm chung với mình theo quy tắc: chỉ cho phép người có thâm niên cao hơn mình và đến đúng giờ leader làm việc mới được vào công ty. Nhiệm vụ của bạn là xác định các lập trình viên Q có được chấp nhận vào làm trong ngày hay không.

Input

Dòng 1 gồm 2 số nguyên H, Q là giờ làm việc và số nhân viên đến công ty trong ngày.

Dòng 2 là 3 số nguyên h, S và E là thâm niên và thời gian làm việc của leader.

Tiếp theo là Q dòng, mỗi dòng gồm 2 số nguyên h_i và t_i là thâm niên và thời gian đến công ty của nhân viên Q_i

Output

In ra Q dòng, mỗi dòng là "YES" hoặc "NO" trả lời cho nhân viên thứ Q có được vào công ty hay không.

Constraints

$$1 \leq H \leq 10^9$$

$$1 \leq Q \leq 10^5$$

$$1 \leq S \leq E \leq H$$

$$1 \leq h_i \leq 10^7$$

$$1 \leq t_i \leq H$$

Ví dụ

Input:	Output:
10 1 5	YES
50 2 6	NO
10 1	NO
10 2	YES
50 5	YES
51 6	
100 10	

17. Bạn có 1 mảng A gồm N số nguyên. Cho 2 hàm $f(X)$ và $g(X)$:
 $f(X)$: là số Z nhỏ nhất sao cho $X < Z \leq N$ và $A[X] < A[Z]$
 $g(X)$: là số Z nhỏ nhất sao cho $X < Z \leq N$ và $A[X] > A[Z]$
 Bạn cần tìm ra mỗi chỉ mục i của mảng $g(f(i))$ với $1 \leq i \leq N$. Nếu số không tồn tại thì in ra -1 , nếu số tồn tại thì in ra $A[g(f(i))]$.

Input

Dòng 1 chứa 1 số nguyên N là kích thước của mảng N . N dòng tiếp theo là từng giá trị của mảng A .

Output

In ra N số nguyên trên 1 dòng là giá trị của $A[g(f(x))]$ hoặc -1 nếu $g(f(x))$ không tồn tại.

Constraint

$$1 \leq N \leq 30000$$

$$0 \leq A[i] \leq 10^{16}$$

Ví dụ

Input:	Output:
8	1 4 4 4 -1 2 -1 -1
3	
7	
1	
7	
8	
4	
5	
2	

Explanation

Next Greater	Next Smaller
3 \rightarrow 7	7 \rightarrow 1
7 \rightarrow 8	8 \rightarrow 4
1 \rightarrow 7	7 \rightarrow 4

7 -> 8	8 -> 4
8 -> -1	-1 -> -1
4 -> 5	5 -> 2
5 -> -1	-1 -> -1
2 -> -1	-1 -> -1

18. Bạn được cho trước 1 stack gồm N số nguyên sao cho số đầu tiên đại diện cho giá trị đỉnh của stack và số cuối cùng là giá trị đáy của stack đó. Bạn cần phải lấy ra (pop) ít nhất L đối tượng trong stack. Bạn có thể chuyển đôi stack thành queue bất cứ lúc nào. Đáy của stack là đầu của queue. Tuy nhiên bạn không thể chuyển đổi queue ngược lại thành stack. Nhiệm vụ của bạn là xóa chính xác K đối tượng của queue sao cho tổng K các đối tượng bị xóa là lớn nhất (K bao gồm các phần tử đã pop).

Input

Dòng đầu tiên bao gồm 3 số N, L và K

Dòng thứ 2 gồm N số nguyên A_i là các giá trị của stack.

Output

In ra tổng tối đa có thể từ K các đối tượng bị xóa.

Constraints

$$1 \leq N \leq 10^5$$

$$1 \leq L \leq N$$

$$1 \leq K \leq N - L$$

$$1 \leq A_i \leq 10^9$$

Ví dụ

Input:

10 5

10 9 1 2 3 4 5 6 7 8

Output:

40

Explanation

Pop 2 đối tượng trong stack: 10, 9.

Chuyển stack thành queue, xóa 3 đối tượng 8, 7, 6.

Tổng maximum là $10 + 9 + 8 + 7 + 6 = 40$

19. Tương tự bài 18

Bạn được cho trước 1 stack gồm N số nguyên sao cho số đầu tiên đại diện cho giá trị đỉnh của stack và số cuối cùng là giá trị đáy của stack đó. Bạn cần phải lấy ra (pop) ít nhất 1 đối tượng trong stack. Bạn có thể chuyển đôi stack thành queue bất cứ lúc nào. Đáy của stack là đầu

của queue. Tuy nhiên bạn không thể chuyển đổi queue ngược lại thành stack. Nhiệm vụ của bạn là xóa chính xác K đối tượng của queue sao cho tổng K các đối tượng bị xóa là lớn nhất (K bao gồm các phần tử đã pop).

2.3 Cây

- Viết chương trình đếm số nút trong cây, tính chiều cao của cây, đếm số nút lá trong cây.
- Viết chương trình tạo cây nhị phân từ bàn phím. Yêu cầu:

Input:

Dòng 1: N số nút của cây

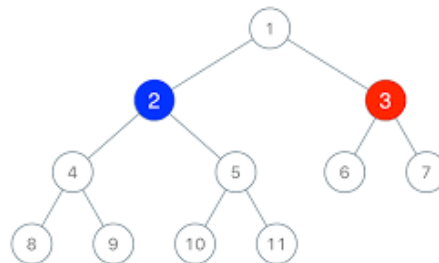
Dòng 2: số nguyên r là giá trị của nút gốc

Dòng 3: 2 số và một ký tự L hoặc R, L đại diện cho node sẽ nằm bên trái và R đại diện cho node sẽ nằm bên phải.

Ouput:

In ra cây nhị phân đã tạo.

Ví dụ:



Input:

10

1 2 L

1 3 R

2 4 L

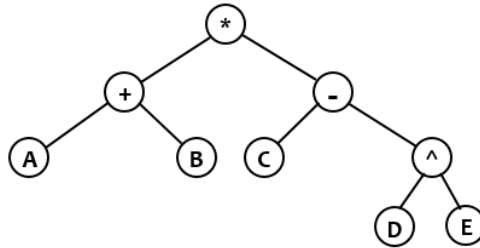
2 5 R

...

- Xử dụng bài 2, nhập từ bàn phím giá trị 1 nút của cây nhị phân vừa tạo, in ra nút đối xứng qua nút gốc với nút vừa nhập, in ra -1 nếu không có nút đối xứng.

4. BITHU - Chiều cao cây biểu thức

Với mọi biểu thức số học gồm 5 phép toán hai ngôi $+$, $-$, $*$, $/$, $^$, ta có thể xây dựng một cây nhị phân tương ứng. Ví dụ với biểu thức hậu tố: $AB + CDE^{\wedge} - *$ thì cây nhị phân tương ứng là như sau:



Yêu cầu: cho một biểu thức hậu tố, hãy xây dựng cây nhị phân tương ứng và tính chiều cao của cây.

Dữ liệu nhập:

- Là chuỗi biểu diễn biểu thức hậu tố gồm số hạng là các chữ cái latin viết hoa và 5 phép toán $+$, $-$, $*$, $/$, $^$. Chiều dài tối đa của chuỗi là 1000. Giữa hai số hạng, hoặc giữa hai phép toán, hoặc giữa số hạng và phép toán, cách nhau một khoảng trắng. Ngoài ra không có ký tự khác. Dữ liệu cho đảm bảo biểu thức hậu tố là hợp lệ.

Dữ liệu xuất:

- Là số nguyên xác định chiều cao của cây nhị phân tương ứng.

Ví dụ

Input

$AB + CDE^{\wedge} - *$

Output

4

5. Cho 1 dãy A gồm N số nguyên, phân loại nó thành 3 loại Good, Bad hoặc Average. A được gọi là Good nếu nó chứa chính xác X số nguyên khác nhau, Bad nếu nó chứa ít hơn X số nguyên khác nhau và Average nếu nó chứa nhiều hơn X số nguyên khác nhau.

Input

Dòng đầu là số nguyên T biểu thị số lần kiểm tra.

Dòng đầu của mỗi lần kiểm tra gồm 2 số N và X .

Dòng thứ 2 của mỗi lần kiểm tra là N số nguyên của mảng A .

Output

In ra câu trả lời của mỗi lần kiểm tra trên 1 dòng.

Constraints

$$1 \leq T \leq 50$$

$$1 \leq X, N \leq 13000$$

$$1 \leq A[i] \leq 10^9$$

Ví dụ

Input:

4

4 1

1 4 2 5

4 2

4 2 1 5

4 3

5 2 4 1

4 4

1 2 4 5

Output:

Average

Average

Average

Good

6. Cây hậu tố

Cho 1 số nguyên K và một chuỗi S có chiều dài N , chúng ta biểu diễn lũy thừa của S là chiều dài của 1 dãy con dài nhất xuất hiện trong S ít nhất K lần.

Cho K và S , tìm lũy thừa của S .

Constraints $1 \leq N \leq 10^6$

$$1 \leq K \leq N$$

Input

Dòng 1 là 2 số K và N . Dòng thứ 2 là chuỗi S .

Output

In ra lũy thừa của chuỗi S với K cho trước.

Ví dụ

Input:

2 11

abrakadabra

Output:

4

Explanation

Chuỗi "abra" xuất hiện trong chuỗi S 2 lần và là chuỗi con có chiều dài lớn nhất trong tất cả các chuỗi con của S xuất hiện ít nhất 2 lần.