**NGÂN HÀNG CÂU HỎI THI THỰC HÀNH PHÒNG MÁY**

**Môn:** Cấu trúc dữ liệu và giải thuật. **Phần:** Ngăn xếp

Phần Stack sẽ có 03 dạng chủ đạo:

1. Vận dụng tư duy kiểm tra dãy ngoặc đúng cơ bản (Mở push, đóng có mở tương ứng là pop)
2. Tính giá trị biểu thức tiền tố, hậu tố
3. Bài toán tìm phần tử đầu tiên bên trái, bên phải phần tử đang xét X với một điều kiện cần xác định trước (≥, >, <, ≤) **và một số bài toán vận dụng (Histogram Hình chữ nhật lớn nhất, …)**

\*Ký hiệu [1] tức là thuộc vào nhóm bài 1

**BÀI 1. DSA\_P010. MIN VÀ MAX [3]**

Cho dãy số A[] có N phần tử là các số nguyên dương không quá 6 chữ số. Người ta tạo ra ma trận C[][] như sau:

* Kích thước của C là N\*N
* Với chỉ số tính từ 1 thì C[i][j] = j \* min (A[i], A[i+1], …, A[i + j - 1]).

Trong đó: 1 ≤ j ≤ n; 1 ≤ i ≤ N – j + 1.

Hãy tìm giá trị lớn nhất của ma trận C.

**Input**

Dòng đầu ghi số N (1 ≤ N ≤ 105).

Dòng tiếp theo ghi N số của dãy A[], các giá trị đều dương và không quá 106.

**Output**

Ghi ra giá trị lớn nhất tính được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  1 4 6 3 2 | 9 |

**BÀI 2. DSA\_P052. ĐÁNH SỐ THỨ TỰ CẶP DẤU NGOẶC [1]**

Cho một biểu thức đúng, tức là các dấu ngoặc đơn đều đầy đủ mở và đóng, đảm bảo đúng thứ tự. Hãy viết chương trình đánh số các cặp dấu ngoặc theo thứ tự xuất hiện, tính từ 1.

Ví dụ với biểu thức (a + (b \*c) ) + (d/e)

ta có thứ tự của các cặp ‘(‘, ‘)’ là 1 2 2 1 3 3

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T (không quá 100).
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một biểu thức số học được đưa vào trên một dòng, độ dài không quá 106.

**Output:** Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  (a + (b \*c) ) + (d/e)  ( ( () ) ( () ) ) | 1  2  2  1  3  3  1 2  3  3  2  4  5  5  4  1 |

**BÀI 3. DSA\_P053. ĐOẠN LIÊN TIẾP [1]**

Cho dãy số A[] có N phần tử. Với mỗi vị trí thứ i trong dãy, hãy tính độ dài của đoạn liên tiếp tính từ i trở về phía trước mà các giá trị đều nhỏ hơn hoặc bằng A[i].

**Input:**Dòng đầu ghi số bộ test (không quá 10). Mỗi test có 2 dòng.

* Dòng đầu tiên gồm 1 số nguyên N (1 ≤ N ≤ 105).
* Dòng tiếp theo gồm N số nguyên A1, A2, …, AN (1 ≤ A[i] ≤ 106).

**Output**

* Với mỗi bộ test, in ra dãy kết quả trên một dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  7  100 80 60 70 60 75 85 | 1 1 1 2 1 4 6 |

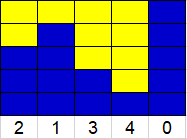
**BÀI 4. DSA\_P054. HÌNH CHỮ NHẬT ĐƠN SẮC [3]**

Cho bảng có kích thước M\*N các ô vuông đơn vị.

Với mỗi dãy số A có N phần tử và các giá trị A[i] thỏa mãn nguyên dương, không vượt quá M, người ta tiến hành tô các ô trong bảng như sau:

* Tô từ trên xuống dưới trong từng cột
* Với cột thứ i, đầu tiên tô A[i] ô màu vàng, tiếp theo là M – A[i] ô màu xanh.

Ví dụ dưới đây là bảng 5\*5 với A[] = {2, 1, 3, 4, 0}.



Hãy tính diện tích hình chữ nhật lớn nhất thỏa mãn:

* Cạnh hình chữ nhật song song với cạnh của bảng
* Tất cả các ô trong hình chữ nhật đều cùng màu.

**Input**

Dòng đầu tiên ghi hai số M và N (0 < N,M ≤106 ).

Dòng thứ 2 ghi N số của dãy A thỏa mãn ràng buộc đề bài.

**Output**

Ghi ra giá trị diện tích tính được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 9  1 3 4 4 5 4 4 3 1 | 21 |

**BÀI 5. DSA\_P055. ĐẾM CẶP SỐ [3] [Mức độ: VDC]**

Cho dãy số nguyên A[]. Với mỗi giá trị A[i], các bạn đã biết ngăn xếp có thể được sử dụng để tìm phần tử đầu tiên bên phải hoặc đầu tiên bên trái lớn hơn giá trị A[i].

Bài toán đặt ra như sau: hãy đếm xem có bao nhiêu cặp (i, j) với i< j thỏa mãn từ vị trí i đến vị trí j không có số nào lớn hơn A[i] hoặc A[j]. Tất nhiên các cặp (i, i + 1) luôn thỏa mãn.

**Input**

Dòng đầu ghi số N là số phần tử của A[] (1 < N < 500.000).

Tiếp theo là N dòng, mỗi dòng ghi 1 số nguyên dương là các phần tử của dãy A, các số đều không quá 9 chữ số.

**Output**

Ghi ra số cặp (i,j) đếm được.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 7  2  4  1  2  2  5  1 | 10 |

***Giải thích test ví dụ:*** *Các cặp thỏa mãn (với chỉ số tính từ 1) là: (1,2); (2,3); (2,4); (2,5); (2,6); (3,4); (4,5); (4,6); (5,6); (6,7).*

**BÀI 3. DSA\_P061. KIỂM TRA CÁC CẶP DẤU NGOẶC [1]**

Cho biểu thức được tạo bởi các cặp “(, )”, “[, ]”, “{, }”.

Hãy cho biết các cặp “(, )”, “[, ]”, “{, }” viết đúng hay không đúng trong biểu thức

**Ví dụ:**

* exp = “[()]{}{[()()]()}” là biểu thức đúng (true)
* exp = “[(])” là sai (false).

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T;
* Những dòng tiếp theo mỗi dòng đưa vào một bộ test. Mỗi bộ test là một biểu thức exp.

**Output:** Đưa ra kết quả mỗi test theo từng dòng.

**Ràng buộc:** T, exp thỏa mãn ràng buộc: 1 ≤ T ≤ 100; 2 ≤ length(exp) ≤ 106.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  [()]{}{[()()]()}  [(]) | true  false |

**BÀI 4. DSA\_P062. SỐ LƯỢNG DẤU NGOẶC VUÔNG LỚN NHẤT [1 – VDC]**

Cho một xâu chỉ gồm các ký tự ‘(‘, ‘)’, ‘[‘, ‘]’. Một dãy ngoặc đúng được định nghĩa như sau:

* Xâu rỗng là một dãy ngoặc đúng
* Nếu A là một dãy ngoặc đúng thì (A), [A] là một dãy ngoặc đúng
* Nếu A và B là 2 dãy ngoặc đúng thì AB là 1 dãy ngoặc đúng

Bài toán đặt ra như sau: Cho một xâu S. Tìm số lượng dấu ‘[‘ lớn nhất có thể trong các xâu con của S (Xâu con là dãy các ký tự liên tiếp) và xâu con cũng phải là một dãy ngoặc đúng

**Input.** Gồm xâu S duy nhất (0 ≤ len(S) ≤ 105)

**Output.** Số lượng dấu ‘[‘ lớn nhất trong xâu con X tìm được

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| [ ( ) ] | 1 |
| [ ( ( ) | 0 |
| [ ( ] ) [ ] | 1 |

**BÀI 5. DSA\_P063. BIỂU THỨC TĂNG GIẢM**

Cho mảng A[] chỉ bao gồm các ký tự I hoặc D. Ký tự I được hiểu là tăng (Increasing) ký tự D được hiểu là giảm (Degreeasin). Sử dụng các số từ 1 đến 9, hãy đưa ra số nhỏ nhất được đoán nhận từ mảng A[]. Chú ý, các số không được phép lặp lại. Dưới đây là một số ví dụ mẫu:

* A[] = “I”               : số tăng nhỏ nhất là 12.
* A[] = “D”              : số giảm nhỏ nhất là 21
* A[] =”DD”            : số giảm nhỏ nhất là 321
* A[] = “DDIDDIID”: số thỏa mãn 321654798

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test là một xâu ID
* T, Length(A) thỏa mãn ràng buộc: 1≤ T ≤100; 1≤ Length(A) ≤9; .

**Output:**

* Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

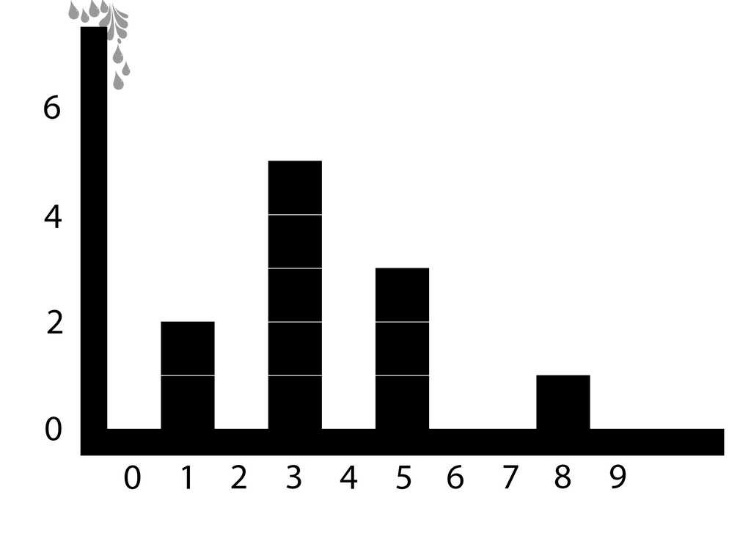
|  |  |
| --- | --- |
| **Input** | **Output** |
| 4  I  D  DD DDIDDIID | 12  21  321  321654798 |

**BÀI 6. DSA\_P064. XÂY ĐẬP [3, Độ khó: 4]**

Trên thượng lưu sông High River, một con đập lớn đang được xây dựng. Do dòng sông chảy xiết, các kĩ sư thiết kế N bức tường chắn nước tại các vị trí L[i], có độ cao tương ứng bằng H[i]. Thiết diện mặt cắt của con đập được biểu diễn như hình vẽ. Mỗi bức tường có độ dày bằng 1 đơn vị.

Do tính phức tạp của con đập, một số bức tường vẫn đang được xây dựng tiếp theo ở phía dưới. Do đó, các kĩ sư cần có hệ thống cảnh báo để xác định giới hạn vùng an toàn cho công việc xây dựng ở phía sau.

Giả sử lưu lượng nước đổ về là K đơn vị thể tích. Các bạn giúp các kĩ sư hãy tính toán xem vị trí bức tường cuối cùng sẽ bị nước tràn qua?



**Input**

* Dòng đầu tiên là số lượng bộ test T (T ≤ 20).
* Mỗi test bắt đầu bởi số lượng bức tường N (N ≤ 105).
* Dòng thứ hai gồm N số nguyên L[] mô tả vị trí của các bức tường (1 ≤ L[i] ≤ 109, L[i] > L[i-1]+1).
* Dòng thứ ba gồm N số nguyên H[] mô tả chiều cao của các bức tường (1 ≤ H[i] ≤ 105).
* Tiếp theo là số lượng truy vấn Q (Q ≤ 105).
* Q dòng tiếp theo, mỗi dòng gồm một số nguyên K (1 ≤ K ≤ 1015).

**Output**

* Với mỗi truy vấn, hãy in ra đáp án trên một dòng. Nếu bức tường thứ nhất không bị vượt qua, in ra 0.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  4  1 3 5 8  2 5 3 1  3  17  3  13 | 3  1  1 |

 Giải thích test:

|  |  |  |
| --- | --- | --- |
| 在这里插入图片描述 | 在这里插入图片描述 | 在这里插入图片描述 |

**BÀI 7. DSA\_P065. LOẠI BỎ DẤU NGOẶC [1, Độ khó: VDC]**

Cho một biểu thức đúng và thỏa mãn:

- Các biến trong biểu thức chỉ chứa các chữ cái viết hoa.

- Các toán tử trong biểu thức là ‘+’ hoặc ‘-’

Hãy loại bỏ các dấu ngoặc thừa mà vẫn giữ nguyên ý nghĩa của biểu thức.

**Input:**

- Dòng đầu tiên chứa số biểu thức M (1<=M<=10).

- M dòng tiếp theo, mỗi dòng là một biểu thức đúng, có thể có các dấu cách tùy ý trong mỗi dòng. Độ dài mỗi dòng (bao gồm cả dấu cách) không quá 255 kí tự.

**Output:**

Với mỗi biểu thức, in ra trên một dòng biểu thức không có các dấu ngoặc thừa.

Chú ý: Thứ tự của các toán hạng trong biểu thức kết quả và biểu thức đầu vào phải giống nhau. Các dấu cách thừa nếu có cũng phải được loại bỏ.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  (A  -  B + C)-(A+   (B–C))  -  (C-   (D- E) )  ((A)-  ((B  )    ))  A-(   B+C) | A-B+C-(A+B-C)-(C-(D-E))  A-B  A-(B+C) |

**BÀI 8. DSA\_P088. KIỂM TRA CÂU VIẾT ĐÚNG QUY TẮC [1]**

Cho một câu văn bản kết thúc với dấu chấm, trong đó chỉ có các chữ cái Tiếng Anh và các dấu ngoặc kiểu ngoặc đơn (,) hoặc ngoặc vuông [,]. Nếu chỉ xét các cặp dấu ngoặc, bạn hãy kiểm tra xem câu văn bản trên có viết đúng quy tắc hay không. Tức là các cặp dấu ngoặc đều viết mở và đóng đẩy đủ, đúng thứ tự.

**Input**.

Dòng đầu ghi số bộ test (không quá 4000).

Mỗi test viết trên một dòng câu văn bản có độ dài không quá 100, luôn kết thúc bởi dấu chấm.

**Output**. Ghi ra YES hoặc NO tùy thuộc kết quả kiểm tra

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  [ graph ] ( binary tree ).  Dynamic programming )( devide and conquer.  ([ (([( [ ] ) ( ) (( ))] )) ]). | YES  NO  YES |

**BÀI 9. DSA\_P091. THAY THẾ DẤU NGOẶC [1]**

Cho một biểu thức toán học với các dấu ngoặc đơn, các toán hạng và các toán tử.

Với mỗi cặp dấu ngoặc đúng, hãy thay thế dấu mở ngoặc thành số 0 và dấu đóng ngoặc thành số 1. Với các vị trí dấu ngoặc không đúng thì thay thế bằng số -1

Ví dụ biểu thức ((a+b)- (c+d) thì sẽ được thay thế thành -10a+b1-0c+d1

**Input**

Dòng đầu ghi số bộ test.

Mỗi bộ test sẽ viết trên một dòng biểu thức có độ dài không quá 1000.

**Output**

Với mỗi bộ test, viết trên một dòng xâu ký tự sau khi đã thay thế dấu ngoặc.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 | -10a1 |
| ((a) | 0a1-1 |
| (a)) | 000abc1100d111-1-1 |
| (((abc))((d))))) |  |

**BÀI 10. DSA\_P096. HÌNH CHỮ NHẬT LỚN NHẤT**

Cho bảng hình chữ nhật có kích thước N x M, các ô trong bảng chỉ nhận giá trị 0 hoặc 1. Nhiệm vụ của bạn là hãy tìm hình chữ nhật chứa toàn số 1, các cạnh song song với trục tọa độ và có diện tích lớn nhất có thể.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test bắt đầu bởi hai số nguyên N và M (N, M ≤ 1000).

N dòng tiếp theo, mỗi dòng gồm M số nguyên mô tả bảng

**Output:**

Với mỗi test, in ra kích thước 2 cạnh của hình chữ nhật (cạnh nhỏ và cạnh lớn). Nếu có nhiều đáp án, in ra đáp án thỏa mãn hiệu của cạnh lớn và cạnh nhỏ là bé nhất có thể.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  4 6  1 1 1 1 1 1  1 1 1 0 0 0  0 0 0 1 0 1  0 1 0 0 1 0  2 3  1 1 1  1 1 0  9 13  0 0 0 0 0 1 0 0 0 0 0 0 0  0 0 1 1 1 1 1 1 1 0 0 0 0  0 0 1 1 1 1 1 1 1 0 0 0 0  0 1 1 1 1 1 1 1 1 1 0 0 0  1 1 1 1 1 1 1 1 1 1 1 0 0  0 1 1 1 1 1 1 1 1 1 0 0 0  0 0 1 1 1 1 1 1 1 0 0 0 0  0 0 1 1 1 1 1 1 1 0 0 0 0  0 0 0 0 1 1 1 0 0 0 0 1 1 | 2 3  2 2  7 7 |

**BÀI 11. DSA\_P114. TỔNG LỚN NHẤT CỦA DÃY CON LIÊN TỤC [3 – Độ khó: 3]**

Cho mảng A[] gồm N phần tử.

Nhiệm vụ của bạn là tìm tổng của tất cả các phần tử lớn nhất của mỗi dãy con liên tục.

**Ví dụ:** Với mảng A[] = {1, 3, 1, 7} ta có kết quả là 42 bằng cách thực hiện như sau:

* Phần tử lớn nhất của dãy con {1}: 1
* Phần tử lớn nhất của dãy con {1, 3}: 3
* Phần tử lớn nhất của dãy con {1, 3, 1}: 3
* Phần tử lớn nhất của dãy con {1, 3, 1, 7}: 7
* Phần tử lớn nhất của dãy con {3}: 3
* Phần tử lớn nhất của dãy con {3, 1}: 3
* Phần tử lớn nhất của dãy con {3, 1, 7}: 7
* Phần tử lớn nhất của dãy con {1}: 1
* Phần tử lớn nhất của dãy con {1, 7}: 7
* Phần tử lớn nhất của dãy con {7}: 7

Kết quả: 1 + 3 + 3 + 7 + 3 + 3 + 7 + 1 + 7 + 7 = 42

**Input:**

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm hai dòng:

Dòng thứ nhất đưa vào số phần tử của mảng N;

Dòng thứ hai đưa vào N phần tử của mảng A[]; các số được viết cách nhau một dấu cách.

* T, N, A[i] thỏa mãn ràng buộc: 1 ≤ T ≤ 100; 1 ≤ N ≤ 106 ; 1 ≤ A[i] ≤ 1000.

**Output:** • Đưa ra kết quả mỗi test theo từng dòng.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  4  1 3 1 7 | 42 |

**Chú ý:** Có 50% số test có 1 ≤ N ≤ 103

**BÀI 12. DSA\_P136. GIẢI MÃ**

Cho một xâu ký tự biểu diễn một đoạn gõ phím, trong đó có 3 ký tự đặc biệt.

* + Kí tự ’-’ biểu diễn phím Backspace: kí tự ở đằng trước con trỏ (nếu có) bị xóa.
  + Kí tự ’<’ biểu diễn phím sang trái: con trỏ sang trái 1 kí tự, nếu có thể.
  + Kí tự ’>’ biểu diễn phím sang phải: con trỏ sang phải 1 kí tự, nếu có thể.

Các kí tự khác là các chữ cái tiếng Anh (in hoa lẫn in thường) hoặc các chữ số sẽ được chèn vào vị trí hiện tại của con trỏ.

Hãy giải mã xâu ký tự và in ra kết quả cuối cùng.

**Input.** Chỉ có 1 xâu ký tự theo mô tả, độ dài không quá 106

**Output.** Ghi ra kết quả giải mã.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| <<PI<T>>Ti- | PTIT |

**BÀI 13. DSA\_P204. TIỀN TỐ [2]**

Biểu thức dưới dạng tiền tố là phép biểu diễn biểu thức trong đó phép toán được đặt trước hai toán hạng. Ví dụ + \* A B / C D tương ứng với biểu thức trung tố A \* B + C / D

Hãy viết chương trình tính toán giá trị của biểu thức tiền tố.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test bắt đầu bởi số nguyên N (N ≤ 1000), là số lượng toán tử và toán hạng.

Dòng tiếp theo là biểu thức tiền tố với N toán tử và toán hạng.

Các toán hạng và toán tử cách nhau bởi một dấu cách.

Các toán hạng có thể có nhiều chữ số, có thể là số âm.

Phép chia chỉ lấy phần nguyên. Ví dụ: 5/3 thì kết quả ghi nhận để tính bước tiếp theo là 1.

Input đảm bảo kết quả phép toán có giá trị tuyệt đối không vượt quá 1018.

**Output:**

Với mỗi test, in ra giá trị biểu thức tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  7  - + 2 \* 3 1 9  7  - + 8 \* 7 5 9  3  + -10 -4 | -4  34  -14 |

**BÀI 14. DSA\_P204. HẬU TỐ[2]**

Biểu thức dưới dạng hậu tố là phép biểu diễn biểu thức trong đó phép toán được đặt sau hai toán hạng. Ví dụ AB+CD-\* (tương ứng với biểu thức trung tố (A+B)\*(C-D).

Hãy viết chương trình tính toán giá trị của biểu thức hậu tố.

**Input:**

Dòng đầu tiên là số lượng bộ test T (T ≤ 20).

Mỗi test bắt đầu bởi số nguyên N (N ≤ 1000), là số lượng toán tử và toán hạng.

Dòng tiếp theo là biểu thức hậu tố với N toán tử và toán hạng. Các toán hạng và toán tử cách nhau bởi một dấu cách.

Các toán hạng có thể có nhiều chữ số, có thể là số âm. Phép chia thực hiện trên số nguyên. Input đảm bảo kết quả phép toán có giá trị tuyệt đối không vượt quá 1018.

**Output:**

Với mỗi test, in ra giá trị biểu thức tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  7  2 3 1 \* + 9 -  7  8 7 5 \* + 9 -  3  -10 -4 + | -4  34  -14 |

**BÀI 15. DSA\_P207. TÍNH TOÁN GIÁ TRỊ CỦA BIỂU THỨC [2]**

Biểu thức dưới dạng hậu tố là phép biểu diễn biểu thức trong đó phép toán được đặt sau hai toán hạng. Ví dụ: AB+CD-\* thì tương ứng với biểu thức trung tố (A+B)\*(C-D).

Biểu thức dưới dạng tiền tố là phép biểu diễn biểu thức trong đó phép toán được đặt trước hai toán hạng. Ví dụ: + \* A B / C D thì tương ứng với biểu thức trung tố A \* B + C / D

Cho một biểu thức ở một trong hai dạng hậu tố hoặc tiền tố. Hãy tính giá trị biểu thức đó.

**Input:**

* Dòng đầu tiên là số lượng bộ test T (T ≤ 50).
* Mỗi test bắt đầu bởi số nguyên N (N ≤ 1000), là số lượng toán tử và toán hạng.
* Dòng tiếp theo là biểu thức hậu tố hoặc tiền tố với N toán tử và toán hạng.
* Các toán hạng và toán tử cách nhau bởi một dấu cách.
* Các toán hạng có thể có nhiều chữ số, có thể là số âm.
* Các toán tử chỉ bao gồm các phép toán cộng, trừ, nhân, chia, trong đó phép chia thực hiện trên số nguyên.
* Input đảm bảo kết quả phép toán có giá trị tuyệt đối không vượt quá 1018

**Output:** Với mỗi test, in ra giá trị biểu thức tìm được.

**Ví dụ:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  7  2 3 1 \* + 9 -  7  - + 8 \* 7 5 9  3  -10 -4 + | -4  34  -14 |

**BÀI 16. DSA\_P220. NHẢY SANG PHẢI [3]**

Cho N cây cột, trong đó cột thứ i có chiều cao là h[i].

Con ếch xuất phát từ một vị trí nào đó và chỉ được phép nhảy đến các cây cột gần nhất có chiều cao lớn hơn cột hiện tại.

Có Q truy vấn, mỗi truy vấn cho biết điểm xuất phát. Hãy tính số bước nhảy mà con ếch có thể thực hiện được từ điểm xuất phát.

**Input**.

* Dòng đầu ghi hai số N và Q (1 ≤ N, Q ≤ 105).
* Dòng thứ hai ghi N số của dãy h[], các giá trị nguyên dương và không quá 109 .
* Q dòng tiếp theo, mỗi dòng ghi một vị trí xuất phát (chỉ số tính từ 1 đến N)

**Output.** Ghi ra Q dòng là số bước nhảy mà con ếch có thể thực hiện tương ứng cho mỗi truy vấn.

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5 5  1 3 4 2 5  1  2  3  4  5 | 3  2  1  1  0 |

**BÀI 17. THU GỌN DÃY SỐ [1]**

Cho dãy số A[] chỉ bao gồm các số nguyên dương. Người ta thu gọn dần dãy số bằng cách loại bỏ các cặp phần tử kề nhau mà có tổng là chẵn. Sau khi cặp phần tử đó bị loại ra thì dãy số được dồn lại. Cứ tiếp tục như vậy cho đến khi không còn cặp phần tử nào kề nhau có tổng chẵn nữa.

Hãy tính xem cuối cùng dãy A[] còn bao nhiêu phần tử.

**Input**

Dòng đâu ghi số N là số phần tử của dãy (1 ≤ N ≤ 105)

Dòng tiếp theo ghi N số của dãy A (1 ≤ A[i] ≤ 100).

**Output**

Ghi ra trên một dòng số phần tử còn lại trong dãy A[].

**Ví dụ**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  2 3 4 5 6 | 5 |
| 10  1 5 5 8 6 4 3 5 9 3 | 2 |

**BÀI 18. RÚT GỌN XÂY KÝ TỰ [1]**

Cho một xâu S. Mỗi bước, bạn được phép xóa đi 2 kí tự liền nhau mà giống nhau. Chẳng hạn xâu “aabcc” có thể trở thành “bcc” hoặc “aab” sau 1 lần xóa.

Hỏi xâu cuối cùng thu được là gì? Nếu xâu rỗng, in ra “Empty String”.

**Input.** Một xâu S chỉ gồm các chữ cái thường, có độ dài không vượt quá 105.

**Output**. In ra đáp án tìm được.

**Ví dụ.**

|  |  |
| --- | --- |
| **Input** | **Output** |
| aaabccddd | abd |
| abba | Empty String |