

[Ngăn xếp]. Bài 1. Thao tác 1

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho một ngăn xếp các số nguyên. Các thao tác gồm 3 lệnh: push, pop và show. Trong đó thao tác push kèm theo một giá trị cần thêm. Hãy viết chương trình ghi ra kết quả của các lệnh show. Trong trường hợp lệnh pop thì ngăn xếp rỗng sẽ không được thực hiện

Input Format

Input: Dòng đầu là số thao tác T; T dòng tiếp theo mỗi dòng chứa một lệnh push, pop hoặc show. Input đảm bảo số lượng phần tử trong stack khi nhiều nhất cũng không vượt quá 100.

Constraints

$1 \leq T \leq 100$;

Output Format

Ghi ra màn hình các phần tử đang có trong stack theo thứ tự lưu trữ mỗi khi gặp lệnh show. Các số viết cách nhau đúng một khoảng trống. Nếu stack rỗng thì in ra EMPTY

Sample Input 0

```
9
show
push 346
push 196
push 946
show
show
push 943
push 265
push 225
```

Sample Output 0

```
EMPTY
346 196 946
346 196 946
```

[Ngăn xếp]. Bài 2. Thao tác 2

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Yêu cầu bạn xây dựng một stack với các truy vấn sau đây: “push x”: Thêm phần tử x vào stack. “top”: In ra phần tử đầu tiên của stack. Nếu stack rỗng, in ra “NONE”. “pop”: Xóa phần tử đầu tiên của stack. Nếu stack rỗng, không làm gì cả.

Input Format

Dòng đầu tiên là số lượng truy vấn T ($T \leq 1000$). Mỗi truy vấn có dạng như trên.

Constraints

N/A

Output Format

Với mỗi truy vấn “top”, hãy in ra phần tử đầu tiên của stack. Nếu stack rỗng, in ra “NONE”.

Sample Input 0

```
8
push 815
push 30
top
push 242
push 502
top
push 876
top
```

Sample Output 0

```
30
502
876
```

[Ngăn xếp]. Bài 3. Valid parentheses

- [Problem](#)
- [Submissions](#)

- [Discussions](#)

Cho một xâu kí tự chỉ bao gồm các kí tự '(', ')', '{', '}', ']', '[', hãy xác định xem các dấu ngoặc trong xâu có cân bằng hay không. Một vài ví dụ về xâu cân bằng : ((())), {{{(O)}}}[], (O)(OO)...

Input Format

Xâu kí tự S trên 1 dòng

Constraints

$1 \leq \text{len}(S) \leq 10000$

Output Format

In ra YES hoặc NO nếu xâu kí tự cân bằng hoặc không.

Sample Input 0

```
(((((O)))))
```

Sample Output 0

```
YES
```

[Ngăn xếp]. Bài 4. Thêm dấu ngoặc

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho một xâu kí tự chỉ bao gồm dấu đóng mở ngoặc tròn. Hãy tìm số lượng dấu đóng mở ngoặc cần thêm tối thiểu để tạo thành một xâu cân bằng, bạn có thể thêm các kí tự đóng mở ngoặc vào bất kì vị trí nào của xâu.

Input Format

Xâu kí tự S trên 1 dòng

Constraints

$1 \leq \text{len}(S) \leq 10000;$

Output Format

In ra số lượng dấu ngoặc cần bổ sung để xâu cân bằng

Sample Input 0

```
)))0
```

Sample Output 0

```
4
```

Sample Input 1

```
((()))
```

Sample Output 1

```
0
```

[Ngăn xếp]. Bài 5. Remove Duplicates

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho một xâu kí tự S chỉ bao gồm kí tự in thường, nhiệm vụ của bạn là xóa các kí tự liên kề giống nhau khỏi xâu S . Ví dụ xâu `aabbccc` sau khi xóa các kí tự liên kề sẽ thành `c`, xâu `abba` sau khi xóa các kí tự liên kề sẽ thành xâu rỗng.

Input Format

Dòng duy nhất chứa xâu kí tự S

Constraints

$1 \leq \text{len}(S) \leq 10000$;

Output Format

In ra xâu S sau khi xóa các kí tự liên kề giống nhau, nếu xâu sau khi xóa rỗng thì in ra `EMPTY`.

Sample Input 0

```
dddbbabcb
```

Sample Output 0

[Ngăn xếp]. Bài 6. Remove Duplicates 2

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho một chuỗi ký tự S chỉ bao gồm ký tự in thường, nhiệm vụ của bạn là xóa k ký tự liên tiếp giống nhau khỏi chuỗi S . Ví dụ chuỗi `aaabbbccccdddeeu` sau khi xóa 3 ký tự liên tiếp giống nhau khỏi chuỗi S sẽ trở thành : `u`

Input Format

Dòng 1 chứa chuỗi ký tự S ; Dòng 2 chứa số nguyên dương k

Constraints

$1 \leq \text{len}(S) \leq 10000$; $2 \leq k \leq \text{len}(S)$;

Output Format

In ra chuỗi S sau khi xóa, in ra `EMPTY` nếu S trở thành chuỗi rỗng

Sample Input 0

```
aaabcccedde  
2
```

Sample Output 0

```
EMPTY
```

Sample Input 1

```
aaaccddda  
3
```

Sample Output 1

```
cca
```

[Ngăn xếp]. Bài 7. Dư thừa dấu ngoặc

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho một biểu thức số học đúng nhưng có thể dư thừa dấu ngoặc, nhiệm vụ của bạn là kiểm tra xem biểu thức này có dư thừa dấu đóng mở ngoặc hay không.

Input Format

Dòng duy nhất chứa biểu thức cần kiểm tra

Constraints

Biểu thức chỉ chứa kí tự đóng mở ngoặc, các chữ cái in thường và các phép toán + - * / không quá 100 kí tự

Output Format

In ra YES nếu biểu thức dư thừa dấu ngoặc, ngược lại in ra NO

Sample Input 0

```
((x+y))
```

Sample Output 0

```
YES
```

[Ngăn xếp]. Bài 8. Lật ngược dấu ngoặc

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho chuỗi S chỉ bao gồm dấu đóng mở ngoặc, hãy tìm số lượng dấu ngoặc tối thiểu cần phải đổi chiều để tạo thành một chuỗi hợp lệ.

Input Format

Dòng duy nhất chứa chuỗi S

Constraints

Xâu S có độ dài là số chẵn và có không quá 1000 kí tự

Output Format

In ra số lượng dấu ngoặc cần phải đổi chiều

Sample Input 0

```
)))())()
```

Sample Output 0

```
2
```

Sample Input 1

```
()()())
```

Sample Output 1

```
0
```

[Ngăn xếp]. Bài 9. Tiền tố - trung tố

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho biểu thức S ở dạng tiền tố, hãy biến đổi S về dạng trung tố và in ra màn hình

Input Format

Dòng duy nhất chứa biểu thức S

Constraints

$1 \leq \text{len}(S) \leq 1000$

Output Format

In ra biểu thức trung tố tương ứng

Sample Input 0

```
-*AB+CD
```

Sample Output 0

```
((A*B)-(C+D))
```

[Ngăn xếp]. Bài 10. Tiền tố - Hậu tố

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho biểu thức S ở dạng tiền tố, hãy biến đổi S về dạng hậu tố và in ra màn hình

Input Format

Dòng duy nhất chứa biểu thức S

Constraints

$1 \leq \text{len}(S) \leq 1000$

Output Format

In ra biểu thức hậu tố tương ứng

Sample Input 0

```
*+AB-UV
```

Sample Output 0

```
AB+UV-*
```

[Ngăn xếp]. Bài 11. Trung tố - hậu tố

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho biểu thức S ở dạng trung tố, hãy biến đổi S về dạng hậu tố và in ra màn hình

Input Format

Dòng duy nhất chứa biểu thức S chỉ chứa các toán tử +, -, *, / và các chữ cái.

Constraints

$1 \leq \text{len}(S) \leq 1000$

Output Format

In ra biểu thức hậu tố tương ứng

Sample Input 0

(X+(Y-Z))

Sample Output 0

XYZ-+

[Ngăn xếp]. Bài 12. Hậu tố - Tiền tố

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho biểu thức S ở dạng hậu, hãy biến đổi S về dạng tiền tố và in ra màn hình

Input Format

Dòng duy nhất chứa biểu thức S

Constraints

$1 \leq \text{len}(S) \leq 1000$

Output Format

In ra biểu thức tiền tố tương ứng

Sample Input 0

AB+UV-/

Sample Output 0

/+AB-UV

[Ngăn xếp]. Bài 13. Hậu tố - Trung tố

- [Problem](#)

- [Submissions](#)
- [Discussions](#)

Cho biểu thức S ở dạng hậu, hãy biến đổi S về dạng trung tố và in ra màn hình

Input Format

Dòng duy nhất chứa biểu thức S

Constraints

$1 \leq \text{len}(S) \leq 1000$

Output Format

In ra biểu thức trung tố tương ứng

Sample Input 0

AB+UV-/

Sample Output 0

$((A+B)/(U-V))$

[Ngăn xếp]. Bài 14. Tính toán giá trị biểu thức tiền tố

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho biểu thức tiền tố S, các số trong biểu thức chỉ là số có 1 chữ số, yêu cầu tính toán giá trị của biểu thức.

Input Format

Dòng duy nhất chứa biểu thức S

Constraints

S có độ dài không quá 1000

Output Format

In ra giá trị của biểu thức S

Sample Input 0

+7/451

Sample Output 0

8

[Ngăn xếp]. Bài 15. Tính toán giá trị biểu thức hậu tố

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho biểu thức hậu tố S, các số trong biểu thức chỉ là số có 1 chữ số, yêu cầu tính toán giá trị của biểu thức.

Input Format

Dòng duy nhất chứa biểu thức S

Constraints

S có độ dài không quá 1000

Output Format

In ra giá trị của biểu thức S

Sample Input 0

316*+9*

Sample Output 0

81

[Ngăn xếp]. Bài 16. Đảo từ

- [Problem](#)
- [Submissions](#)

- [Discussions](#)

Cho một đoạn văn có không quá 10000 từ, nhiệm vụ của bạn là in ra các từ trong đoạn văn theo thứ tự ngược.

Input Format

Gồm nhiều dòng, mỗi dòng gồm nhiều từ.

Constraints

Có không quá 10000 từ trong input

Output Format

In ra các từ trong đoạn văn theo thứ tự ngược

Sample Input 0

```
C
special word C++ word 28tech
word programming
word programming C
Python C++
```

Sample Output 0

```
C++ Python C programming word programming word 28tech word C++ word special C
```

[Ngăn xếp]. Bài 17. Next greater element (NGE)

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng $A[]$ gồm N phần tử, nhiệm vụ của bạn là tìm phần tử đầu tiên bên phải lớn hơn của mọi phần tử $A[i]$ trong mảng, nếu phần tử nào không có phần tử bên phải lớn hơn thì in ra -1.

Input Format

Dòng đầu tiên chứa N : Số lượng phần tử trong mảng. Dòng thứ 2 chứa N số nguyên, mỗi số cách nhau một khoảng trắng.

Constraints

$1 \leq N \leq 10^6$; $1 \leq A[i] \leq 10^6$;

Output Format

In ra đáp án của bài toán trên 1 dòng.

Sample Input 0

```
8
19 11 11 15 13 18 19 18
```

Sample Output 0

```
-1 15 15 18 18 19 -1 -1
```

[Ngăn xếp]. Bài 18. Phần tử nhỏ hơn bên phải

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho mảng $A[]$ có n phần tử, nhiệm vụ của bạn là đối với mỗi phần tử $A[i]$ trong mảng phải tìm ra phần tử nhỏ hơn đầu tiên nằm bên phải của phần tử lớn hơn đầu tiên nằm bên phải của $A[i]$. Đối với những phần tử không có phần tử lớn hơn bên phải thì in ra -1. Ví dụ $A[] = \{1, 5, 3, 3, 4, 1\}$, $1 \rightarrow 5 \rightarrow 3$, $5 \rightarrow -1 \rightarrow -1$, $3 \rightarrow 4 \rightarrow -1$, $3 \rightarrow 4 \rightarrow -1$, $4 \rightarrow -1 \rightarrow -1$, $1 \rightarrow -1 \rightarrow -1$.

Input Format

Dòng đầu tiên chứa N : Số lượng phần tử trong mảng. Dòng thứ 2 chứa N số nguyên, mỗi số cách nhau một khoảng trắng.

Constraints

$1 \leq N \leq 10^6$; $1 \leq A[i] \leq 10^6$;

Output Format

In ra đáp án của bài toán trên 1 dòng.

Sample Input 0

```
5
```

11 18 12 18 10

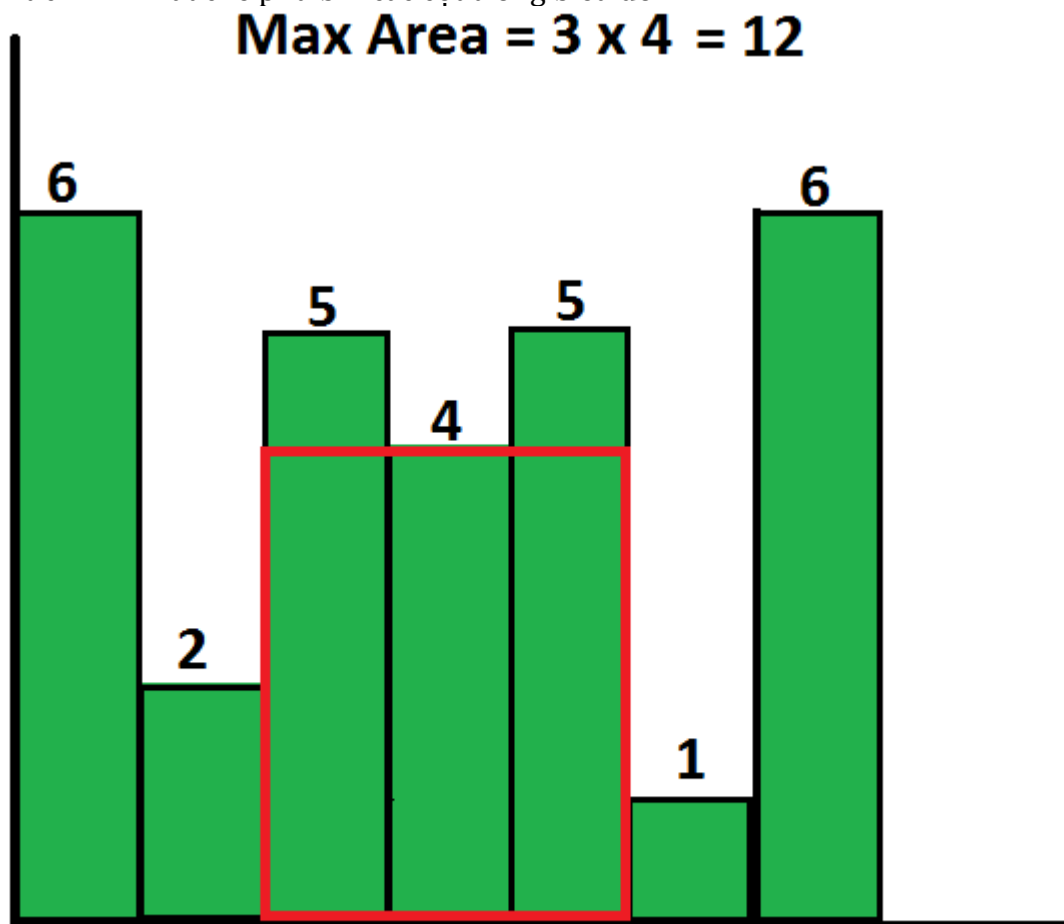
Sample Output 0

12 -1 10 -1 -1

[Ngăn xếp]. Bài 19. Largest Rectangular Area in a Histogram

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho một biểu đồ gồm các cột với chiều cao $H[i]$, nhiệm vụ của bạn là tìm hình chữ nhật có diện tích lớn nhất che phủ bởi các cột trong biểu đồ.



Input Format

Dòng đầu tiên là N : số lượng cột trong biểu đồ. Dòng thứ 2 gồm N số là chiều cao các cột trong biểu đồ.

Constraints

$1 \leq N \leq 10^5$; $1 \leq H[i] \leq 10^6$

Output Format

In ra diện tích lớn nhất tìm được.

Sample Input 0

```
7
6 2 5 4 5 1 6
```

Sample Output 0

```
12
```

[Ngăn xếp]. Bài 20. Hình chữ nhật 0-1

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho một bảng kích thước $N \times M$, được chia thành lưới ô vuông đơn vị N dòng M cột. Trên các ô của bảng ghi số 0 hoặc 1. Nhiệm vụ của bạn là tìm hình chữ nhật gồm các ô của bảng thoả mãn các điều kiện sau: - Hình chữ nhật đó chỉ gồm các số 1. - Cạnh hình chữ nhật song song với cạnh bảng. - Diện tích hình chữ nhật là lớn nhất.

Input Format

Dòng đầu tiên là N và M; N dòng tiếp theo mỗi dòng gồm M số 0 hoặc 1.

Constraints

$1 \leq N, M \leq 500$; Các phần tử trong bảng là 0 hoặc 1.

Output Format

In ra đáp án của bài toán

Sample Input 0

```
6 6
1 0 0 1 1 0
0 1 0 0 0 0
0 0 1 0 0 1
0 0 1 1 0 1
0 1 1 0 0 1
0 0 1 1 0 1
```

Sample Output 0

```
4
```

[Hàng Đợi]. Bài 1. Thao tác cơ bản

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Ban đầu cho một queue rỗng. Bạn cần thực hiện các truy vấn sau: 1. Trả về kích thước của queue 2. Kiểm tra xem queue có rỗng không, nếu có in ra "YES", nếu không in ra "NO". 3. Cho một số nguyên và đẩy số nguyên này vào cuối queue. 4. Loại bỏ phần tử ở đầu queue nếu queue không rỗng, nếu rỗng không cần thực hiện. 5. Trả về phần tử ở đầu queue, nếu queue rỗng in ra -1. 6. Trả về phần tử ở cuối queue, nếu queue rỗng in ra -1.

Input Format

Dòng đầu tiên chứa số nguyên n - lượng truy vấn ($1 \leq n \leq 1000$) N dòng tiếp theo, mỗi dòng sẽ ghi loại truy vấn như trên, với truy vấn loại 3 sẽ có thêm một số nguyên, không quá 10^6 .

Constraints

N/A

Output Format

In ra kết quả của các truy vấn trên từng dòng

Sample Input 0

```
7
2
6
2
5
6
3 495
```


Sample Output 0

```
YES
-1
YES
-1
-1
495
```

[Hàng Đợi]. Bài 2. Thao tác cơ bản 2

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Yêu cầu bạn xây dựng một queue với các truy vấn sau đây: “PUSH x”: Thêm phần tử x vào cuối của queue ($0 \leq x \leq 1000$). “PRINTFRONT”: In ra phần tử đầu tiên của queue. Nếu queue rỗng, in ra “NONE”. “POP”: Xóa phần tử ở đầu của queue. Nếu queue rỗng, không làm gì cả.

Input Format

Dòng đầu tiên là số lượng truy vấn Q ($Q \leq 100000$). Mỗi truy vấn có dạng như trên.

Constraints

N/A

Output Format

Với mỗi truy vấn “PRINT”, hãy in ra phần tử đầu tiên của queue. Nếu queue rỗng, in ra “NONE”.

Sample Input 0

```
10
PUSH 681
POP
PUSH 405
PUSH 983
PUSH 147
POP
PUSH 417
PUSH 918
```

PRINTFRONT
POP

Sample Output 0

983

[Hàng Đợi]. Bài 3. Thao tác cơ bản 3

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Yêu cầu bạn xây dựng một hàng đợi hai đầu với các truy vấn sau đây: - “PUSHFRONT x”: Thêm phần tử x vào đầu của dequeue ($0 \leq x \leq 1000$). - “PRINTFRONT”: In ra phần tử đầu tiên của dequeue. Nếu dequeue rỗng, in ra “NONE”. - “POPFRONT”: Xóa phần tử đầu của dequeue. Nếu dequeue rỗng, không làm gì cả. - “PUSHBACK x”: Thêm phần tử x vào cuối của dequeue ($0 \leq x \leq 1000$). - “PRINTBACK”: In ra phần tử cuối của dequeue. Nếu dequeue rỗng, in ra “NONE”. - “POPBACK”: Xóa phần tử cuối của dequeue. Nếu dequeue rỗng, không làm gì cả.

Input Format

Dòng đầu tiên là số lượng truy vấn Q ($Q \leq 100000$). Mỗi truy vấn có dạng như trên.

Constraints

N/A

Output Format

Với mỗi truy vấn “PRINTFRONT” và “PRINTBACK”, hãy in ra kết quả trên một dòng.

Sample Input 0

```
10
PUSHBACK 327
PUSHFRONT 354
PRINTBACK
PUSHBACK 999
POPBACK
POPBACK
POPBACK
PRINTBACK
PUSHBACK 549
```

Sample Output 0

327
NONE

[Hàng Đợi]. Bài 4. Số nhị phân

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho số tự nhiên n . Hãy in ra tất cả các số nhị phân từ 1 đến n . Chú ý không in thừa các bit 0 ở đầu.

Input Format

Một số tự nhiên n được ghi trên một dòng.

Constraints

$1 \leq n \leq 100000$;

Output Format

In ra các số nhị phân từ 1 tới n .

Sample Input 0

7

Sample Output 0

1 10 11 100 101 110 111

[Hàng Đợi]. Bài 5. Bội số

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho số tự nhiên N . Hãy tìm số nguyên dương X nhỏ nhất được tạo bởi số 8 và số 0 chia hết cho N . Ví dụ với $N = 5$ ta sẽ tìm ra $X = 80$.

Input Format

Dòng đầu tiên ghi lại số lượng test T ($T \leq 100$). Những dòng kế tiếp mỗi dòng ghi lại một test. Mỗi test là một số tự nhiên N được ghi trên một dòng ($N \leq 200$).

Constraints

N/A

Output Format

Đưa ra theo từng dòng số X nhỏ nhất chia hết cho N tìm được

Sample Input 0

```
6
61
79
45
129
5
176
```

Sample Output 0

```
800808
80080088
8888888880
800800008
80
880
```

[Hàng Đợi]. Bài 6. Số BDN SPOJ

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Ta gọi số nguyên dương K là một số BDN nếu các chữ số trong K chỉ bao gồm các 0 hoặc 1 có nghĩa. Ví dụ số $K = 1, 10, 101$. Cho số tự nhiên N ($N \leq 2^{63}$). Hãy cho biết có bao nhiêu số BDN nhỏ hơn N . Ví dụ $N=100$ ta có 3 số BDN bao gồm các số: 1, 10, 11.

Input Format

Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test; T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một số tự nhiên N .

Constraints

$1 \leq T \leq 100$; $1 \leq N \leq 2^{63}$

Output Format

Đưa ra kết quả mỗi test theo từng dòng.

Sample Input 0

```
12
27655
16862
30017
12246
26510
6386
3668
6127
27269
24195
20859
5416
```

Sample Output 0

```
31
31
31
31
31
15
15
15
31
31
31
15
```

[Hàng Đợi]. Bài 7. SỐ BDN 2

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Ta gọi số nguyên dương K là một số BDN nếu các chữ số trong K chỉ bao gồm các 0 hoặc 1 có nghĩa. Ví dụ số $K = 101$ là số BDN, $k=102$ không phải là số BDN. Số BDN của N là số $P = M'N$ sao cho P là số BDN. Cho số tự nhiên N , hãy tìm số BDN nhỏ nhất của N . Ví dụ. Với

$N=2$, ta tìm được số BDN của N là $P = 5^2=10$. $N = 17$ ta tìm được số BDN của 17 là $P = 653^17=11101$.

Input Format

• Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test; • T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một số tự nhiên N .

Constraints

$1 \leq T \leq 100$; $1 \leq N \leq 500$;

Output Format

Đưa ra kết quả mỗi test theo từng dòng.

Sample Input 0

```
14
19
18
9
10
27
23
14
22
40
38
6
13
30
43
```

Sample Output 0

```
11001
1111111110
111111111
10
110111111
110101
10010
110
1000
110010
1110
1001
1110
1101101
```

[Hàng Đợi]. Bài 8. Số lộc phát

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Một số được gọi là lộc phát nếu chỉ có 2 chữ số 6 và 8. Cho số tự nhiên N . Hãy liệt kê các số lộc phát có không quá N chữ số.

Input Format

- Dòng đầu tiên ghi lại số tự nhiên T là số lượng bộ test ($T < 10$);
- T dòng kế tiếp mỗi dòng ghi số N

Constraints

$1 < N \leq 18$.

Output Format

- In ra đáp án theo thứ tự giảm dần.

Sample Input 0

```
2
2
3
```

Sample Output 0

```
88 86 68 66 8 6
888 886 868 866 688 686 668 666 88 86 68 66 8 6
```

[Hàng Đợi]. Bài 9. Số lộc phát 2

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Một số được gọi là lộc phát nếu chỉ có 2 chữ số 6 và 8. Cho số tự nhiên N . Hãy liệt kê các số lộc phát có không quá N chữ số.

Input Format

- Dòng đầu tiên ghi lại số tự nhiên T là số lượng bộ test ($T < 10$);
- T dòng kế tiếp mỗi dòng ghi số N (1

Constraints

N/A

Output Format

Dòng đầu tiên là số lượng số lộc phát tìm được. Dòng thứ hai in đáp án theo thứ tự tăng dần.

Sample Input 0

```
2
2
3
```

Sample Output 0

```
6
6 8 66 68 86 88
14
6 8 66 68 86 88 666 668 686 688 866 868 886 888
```

[Hàng Đợi]. Bài 10. Số lộc phát 3

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Một số được gọi là lộc phát nếu chỉ có 2 chữ số 6 và 8. Cho số tự nhiên N . Hãy liệt kê các số lộc phát có không quá N chữ số.

Input Format

- Dòng đầu tiên ghi lại số tự nhiên T là số lượng bộ test ($T < 20$);
- T dòng kế tiếp mỗi dòng ghi số N (1

Constraints

N/A

Output Format

Dòng đầu tiên là số lượng số lộc phát tìm được. Dòng thứ hai in ra đáp án theo thứ tự giảm dần.

Sample Input 0

```
2
2
3
```

Sample Output 0

```
6
88 86 68 66 8 6
14
888 886 868 866 688 686 668 666 88 86 68 66 8 6
```

[Hàng Đợi]. Bài 11. Giá trị xâu kí tự

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho xâu ký tự $S[]$ bao gồm các ký tự in hoa $[A, B, \dots, Z]$. Ta định nghĩa giá trị của xâu $S[]$ là tổng bình phương số lần xuất hiện mỗi ký tự trong xâu. Ví dụ với xâu $S[] = \text{"AAABBCD"}$ ta có $F(S) = 3^2 + 2^2 + 1^2 + 1^2 = 15$. Hãy tìm giá trị nhỏ nhất của xâu $S[]$ sau khi loại bỏ K ký tự trong xâu.

Input Format

Dòng đầu tiên đưa vào số lượng test T . Mỗi test được tổ chức thành 2 dòng. Dòng thứ nhất ghi lại số K . Dòng thứ 2 ghi lại xâu ký tự $S[]$ có độ dài không vượt quá 10^6 .

Constraints

$T \leq 100$; $1 \leq K \leq 10^6$; $1 \leq \text{len}(S) \leq 10^6$; Xâu S chỉ bao gồm các ký tự in hoa hoặc in thường.

Output Format

Đưa ra giá trị nhỏ nhất của mỗi test theo từng dòng.

Sample Input 0

```
2
0
ABCC
1
ABCC
```

Sample Output 0

[Hàng Đợi]. Bài 12. Minimum operation

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho hai số nguyên dương S và T và hai thao tác (a), (b) dưới đây: Thao tác (a): Trừ S đi 1 ($S = S - 1$); Thao tác (b): Nhân S với 2 ($S = S * 2$); Hãy dịch chuyển S thành T sao cho số lần thực hiện các thao tác (a), (b) là ít nhất. Ví dụ với $S = 2, T = 5$ thì số các bước ít nhất để dịch chuyển S thành T thông qua 4 thao tác sau: Thao tác (a): $2 * 2 = 4$; Thao tác (b): $4 - 1 = 3$; Thao tác (a): $3 * 2 = 6$; Thao tác (b): $6 - 1 = 5$;

Input Format

- Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test;
- T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một bộ đôi S và T .

Constraints

$$1 \leq S, T \leq 10000$$

Output Format

Đưa ra kết quả mỗi test theo từng dòng.

Sample Input 0

```
6
17 16
16 10
13 17
13 10
17 17
16 19
```

Sample Output 0

```
1
6
6
3
0
```

[Hàng Đợi]. Bài 13. Minimum operation 2

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho số tự nhiên N và hai phép biến đổi (a), (b) dưới đây. • Thao tác (a): Trừ N đi 1 ($N = N - 1$). Ví dụ $N = 17$, thao tác (a) biến đổi $N = N - 1 = 16$. • Thao tác (b): $N = \max(u, v)$ nếu $u * v = N$ ($u > 1, v > 1$). Ví dụ $N = 16$, thao tác (b) có thể biến đổi $N = \max(2, 8) = 8$ hoặc $N = \max(4, 4) = 4$. Chỉ được phép sử dụng hai thao tác (a) hoặc (b), hãy biến đổi N thành 1 sao số các thao tác (a), (b) được thực hiện ít nhất. Ví dụ với $N = 17$, số các phép (a), (b) nhỏ nhất biến đổi N thành 1 là 4 bước như sau: - Thao tác (a): $N = N - 1 = 17 - 1 = 16$. - Thao tác (b): $16 = \max(4, 4) = 4$. - Thao tác (b): $4 = \max(2, 2) = 2$. - Thao tác (a): $2 = 2 - 1 = 1$.

Input Format

• Dòng đầu tiên ghi lại số tự nhiên T là số lượng Test; • T dòng kế tiếp mỗi dòng ghi lại một bộ Test. Mỗi test là một số N

Constraints

$1 \leq T \leq 100$; $1 \leq N < 10^9$;

Output Format

• Đưa ra kết quả mỗi test theo từng dòng

Sample Input 0

```
7
896
99
841
880
227
870
291
```

Sample Output 0

```
5
6
6
```

5
7
5
6

[Hàng Đợi]. Bài 14. Chữ số khác biệt

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho hai số nguyên L, R . Hãy đưa ra số các số K nguyên dương trong khoảng $[L, R]$ thỏa mãn điều kiện: • Tất cả các chữ số của K đều khác nhau. • Tất cả các chữ số của K đều nhỏ hơn hoặc bằng 5. Ví dụ với $L = 4, R = 13$ ta có 5 số thỏa mãn yêu cầu là 4, 5, 10, 12, 13,

Input Format

• Dòng đầu tiên đưa vào số lượng test T . • Dòng tiếp theo đưa vào các bộ test. Mỗi bộ test được là một cặp L, R được viết trên một dòng.

Constraints

• T, L, R thỏa mãn ràng buộc: $1 \leq T \leq 100; 0 \leq L \leq R \leq 10^5$.

Output Format

• Đưa ra kết quả mỗi test theo từng dòng.

Sample Input 0

```
8
9 41
91 163
28 59
12 38
72 127
48 68
50 101
22 100
```

Sample Output 0

```
17
20
15
```

14
8
5
5
18

[Hàng Đợi]. Bài 15. Biến đổi hợp lệ

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho cặp số S và T là các số nguyên tố có 4 chữ số (Ví dụ S = 1033, T = 8197 là các số nguyên tố có 4 chữ số). Hãy viết chương trình tìm cách dịch chuyển S thành T thỏa mãn đồng thời những điều kiện dưới đây: a. Mỗi phép dịch chuyển chỉ được phép thay đổi một chữ số của số ở bước trước đó (ví dụ nếu S=1033 thì phép dịch chuyển S thành 1733 là hợp lệ); b. Số nhận được cũng là một số nguyên tố có 4 chữ số (ví dụ nếu S=1033 thì phép dịch chuyển S thành 1833 là không hợp lệ, và S dịch chuyển thành 1733 là hợp lệ); c. Số các bước dịch chuyển là ít nhất. Ví dụ số các phép dịch chuyển ít nhất để S = 1033 thành T = 8179 là 6 bao gồm các phép dịch chuyển như sau:
8179 8779 3779 3739 3733 1733 1033.

Input Format

- Dòng đầu tiên đưa vào số lượng test T ($T \leq 100$)
- Những dòng kế tiếp mỗi dòng đưa vào một test. Mỗi test là một bộ đôi S, T.

Constraints

N/A

Output Format

- Đưa ra kết quả mỗi test theo từng dòng.

Sample Input 0

8
3727 4649
3989 5347
1249 5821
9857 4603
4903 7949
3533 9337
7949 1087
8867 6277

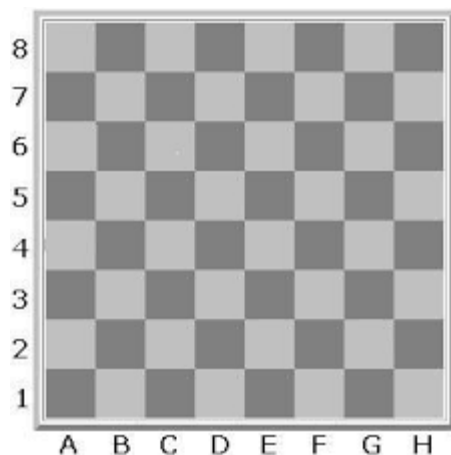
Sample Output 0

```
5
5
5
6
4
4
5
4
```

[Hàng Đợi]. Bài 16. Knight SPOJ

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho một quân mã trên bàn cờ vua tại vị trí ST. Nhiệm vụ của bạn là hãy tìm số bước di chuyển ít nhất để đưa quân mã tới vị trí EN.



Input Format

Dòng đầu tiên là số lượng bộ test T ($T \leq 20$). Mỗi test gồm 2 xâu dạng "xy" và "uv", trong đó x, y là kí tự trong "abcdefgh" còn u, v là số thuộc 1, 2, 3, 4, 5, 6, 7, 8.

Constraints

N/A

Output Format

Với mỗi test, in ra đáp án tìm được trên một dòng. Nếu không thể tìm được đường in ra -1.

Sample Input 0

```
5
b2 c6
h4 b3
g1 b2
e2 c3
f6 b2
```

Sample Output 0

```
3
3
4
1
4
```

[Hàng Đợi]. Bài 17. Di chuyển trong ma trận

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Cho ma trận $A[M][N]$. Nhiệm vụ của bạn hãy tìm số bước đi ít nhất dịch chuyển từ vị trí $A[1][1]$ đến vị trí $A[M][N]$. Biết mỗi bước đi ta chỉ được phép dịch chuyển đến vị trí $A[i][j+A[i][j]]$ hoặc vị trí $A[i+A[i][j]][j]$ bên trong ma trận.

Input Format

- Dòng đầu tiên đưa vào số lượng test T .
- Dòng tiếp theo đưa vào các bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất là hai số M, N ; phần thứ hai là các phần tử của ma trận $A[][]$; các số được viết cách nhau một vài khoảng trống.

Constraints

$T, M, N, A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq M, N, A[i][j] \leq 10^3$.

Output Format

Đưa ra kết quả mỗi test theo từng dòng. In ra -1 nếu không tìm được đáp án.

Sample Input 0

```
1
3 3
2 1 2
```

1 1 1
1 1 1

Sample Output 0

2

[Hàng Đợi]. Bài 18. HEX

- [Problem](#)
- [Submissions](#)
- [Discussions](#)

Có một chiếc bảng hình chữ nhật với 6 miếng ghép, trên mỗi miếng ghép được điền một số nguyên trong khoảng từ 1 đến 6. Tại mỗi bước, chọn một hình vuông (bên trái hoặc bên phải), rồi quay theo chiều kim đồng hồ.

4	1	3
5	2	6

1	2	3
4	5	6

1	5	2
4	6	3

Yêu cầu: Cho một trạng thái của bảng, hãy tính số phép biến đổi ít nhất để đưa bảng đến trạng thái đích. Link submit : <https://www.spoj.com/PTIT/problems/SSAM319H/>

```
#include <bits/stdc++.h>
using namespace std;
```

```
vector<int> trai(vector<int> a){
    vector<int> res(6);
    res[0] = a[0]; res[1] = a[4]; res[2] = a[1];
    res[3] = a[3]; res[4] = a[5]; res[5] = a[2];
    return res;
}
```

```
vector<int> phai(vector<int> a){
    vector<int> res(6);
    res[0] = a[3]; res[1] = a[0]; res[2] = a[2];
    res[3] = a[4]; res[4] = a[1]; res[5] = a[5];
    return res;
}
```

```
bool check(vector<int> a, vector<int> b){
```



```

for(int i = 0; i < 6; i++){
    if(a[i] != b[i]) return false;
}
return true;
}

int bfs(vector<int> a, vector<int> b){
    queue<pair<vector<int>, int>> q;
    q.push({a, 0});
    set<vector<int>> s; s.insert(a);
    while(!q.empty()){
        pair<vector<int>,int> top = q.front(); q.pop();
        if(check(top.first, b)){
            return top.second;
        }
        vector<int> left = trai(top.first);
        if(!s.count(left)){
            q.push({left, top.second + 1});
            s.insert(left);
        }
        vector<int> right = phai(top.first);
        if(!s.count(right)){
            q.push({right, top.second + 1});
            s.insert(right);
        }
    }
}

int main(){
    int t = 1;
    while(t--){
        vector<int> a(6), b(6);
        for(int i = 0; i < 6; i++) cin >> a[i];
        for(int i = 0; i < 6; i++) cin >> b[i];
        cout << bfs(a, b) << endl;
    }
    return 0;
}

```

Input Format

Dòng đầu tiên chứa 6 số là trạng thái bảng ban đầu (thứ tự từ trái qua phải, dòng 1 tới dòng 2). Dòng thứ hai chứa 6 số là trạng thái bảng đích (thứ tự từ trái qua phải, dòng 1 tới dòng 2).

Constraints

N/A

Output Format

In ra một số nguyên là đáp số của bài toán.

Nối dây 1

- [Problem](#)
- [Submissions](#)
- [Leaderboard](#)
- [Discussions](#)

Cho N sợi dây với độ dài khác nhau được lưu trong mảng A[]. Nhiệm vụ của bạn là nối N sợi dây thành một sợi sao cho tổng chi phí nối dây là nhỏ nhất. Biết chi phí nối sợi dây thứ i và sợi dây thứ j là tổng độ dài hai sợi dây A[i] và A[j] Gợi ý : Sử dụng hàng đợi ưu tiên, priority_queue

Tutorial : https://www.youtube.com/watch?v=DRcAJNhtwbY&t=559s&ab_channel=andrew2804

Input Format

Dòng thứ nhất đưa vào số lượng sợi dây N; Dòng tiếp theo đưa vào N số A[i] là độ dài của các sợi dây; Các số được viết cách nhau một vài khoảng trống.

Constraints

$1 \leq N \leq 10^6$; $0 \leq A[i] \leq 10^6$.

Output Format

In ra chi phí nối dây tối thiểu.

Sample Input 0

```
4
4 3 2 6
```

Sample Output 0

```
29
```

Sample Input 1

```
5
4 2 7 6 9
```

Sample Output 1

```
62
```

Nối dây 2

- [Problem](#)
- [Submissions](#)
- [Leaderboard](#)
- [Discussions](#)

Cho N sợi dây với độ dài khác nhau được lưu trong mảng A[]. Nhiệm vụ của bạn là nối N sợi dây thành một sợi sao cho tổng chi phí nối dây là lớn nhất. Biết chi phí nối sợi dây thứ i và sợi dây thứ j là tổng độ dài hai sợi dây A[i] và A[j] Gợi ý : Sử dụng hàng đợi ưu tiên, priority_queue

Tutorial : https://www.youtube.com/watch?v=DRcAJNhtwbY&t=559s&ab_channel=andrew2804

Input Format

Dòng thứ nhất đưa vào số lượng sợi dây N; Dòng tiếp theo đưa vào N số A[i] là độ dài của các sợi dây; Các số được viết cách nhau một vài khoảng trống.

Constraints

$1 \leq N \leq 10^6$; $0 \leq A[i] \leq 10^5$.

Output Format

In ra chi phí nối dây lớn nhất.

Sample Input 0

```
5
1 5 3 4 2
```

Sample Output 0

```
50
```

Explanation 0

Giải thích, (1, 5, 3, 4, 2) -> (1, 9, 3, 2) -> (1, 12, 2) -> (1, 14) -> 15 Chi phí = 4 + 5 + 9 + 3 + 12 + 2 + 14 + 1 = 50

Nối dây 3.

- [Problem](#)
- [Submissions](#)

- [Leaderboard](#)
- [Discussions](#)

Cho N sợi dây với độ dài khác nhau được lưu trong mảng A[]. Nhiệm vụ của bạn là nối N sợi dây thành một sợi sao cho tổng chi phí nối dây là nhỏ nhất. Biết chi phí nối sợi dây thứ i và sợi dây thứ j là tổng độ dài hai sợi dây A[i] và A[j] Gợi ý : Sử dụng hàng đợi ưu tiên, priority_queue

Tutorial : https://www.youtube.com/watch?v=DRcAJNhtwbY&t=559s&ab_channel=andrew2804

Input Format

Dòng thứ nhất đưa vào số lượng sợi dây N; Dòng tiếp theo đưa vào N số A[i] là độ dài của các sợi dây; Các số được viết cách nhau một vài khoảng trống.

Constraints

$1 \leq N \leq 10^6$; $1 \leq A[i] \leq 10^9$.

Output Format

In ra chi phí nhỏ nhất lấy dư với $(10^9 + 7)$.

Sample Input 0

```
5
4 2 7 6 9
```

Sample Output 0

```
62
```