

REARRANG ARRAY PROBLEMS

1. **Re-arrang Array 1.** Cho mảng $A[]$ gồm n phần tử. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao cho $A[i] = i$. Nếu phần tử $A[j]$ của có giá trị khác j , hãy đưa ghi vào -1. Ví dụ với mảng $A[] = \{-1, -1, 6, 1, 9, 3, 2, -1, 4, -1\}$ ta có kết quả $A[] = \{-1, 1, 2, 3, 4, -1, 6, -1, -1, 9\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^5$; $-1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	-1 1 2 3 4 -1 6 -1 -1 9
10	0 1 -1 3 -1 -1
-1 -1 6 1 9 3 2 -1 4 -1	
6	
0 -3 1 -2 3 -4	

2. **Re-arrang Array 2.** Cho mảng $A[]$ gồm n phần tử. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao cho các số 0 để ở cuối cùng, các phần tử khác không được bảo toàn thứ tự trước sau. Ví dụ với mảng $A[] = \{1, 2, 0, 0, 0, 3, 6\}$ ta có kết quả $A[] = \{1, 2, 3, 6, 0, 0, 0\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $1 \leq A[i] \leq 10^{18}$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 3 6 0 0 0
7	1 2 3 0 0 0
1 2 0 0 0 3 6	
6	
0 1 0 2 0 3	

3. **Re-arrang Array 3.** Cho mảng $A[]$ gồm n số nguyên dương và số k . Sử dụng ít nhất các phép đổi chỗ, hãy sắp đặt lại các phần tử của mảng sao cho các số nhỏ hơn hoặc bằng k đứng cạnh nhau. Ví dụ với mảng $A[] = \{2, 1, 5, 6, 3\}$ ta chỉ cần thực hiện 1 phép đổi chỗ để có mảng $A[] = \{2, 1, 3, 6, 5\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$ và số k ; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, k, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^7$; $1 \leq A[i], k \leq 10^7$;

Output:

- Đưa ra số lượng các phép đổi chỗ ít nhất để các số nhỏ hơn hoặc bằng k đứng cạnh nhau.

Input:	Output:
2	1
5 3	2
2 1 5 6 3	
7 5	
2 7 9 5 8 7 4	

4. **Re-arrang Array 4.** Cho mảng $A[]$ gồm n số nguyên dương. Nhiệm vụ của bạn là hãy sắp xếp lại các phần tử của mảng sao cho $A[i] \geq A[i-1]$ nếu i chẵn, $A[i] \leq A[i-1]$ nếu i lẻ. Ví dụ với mảng $A[] = \{1, 2, 2, 1\}$ ta được mảng được sắp $A[] = \{1, 2, 1, 2\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^3$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	1 2 1 2
4	1 3 2
1 2 2 1	
3	
1 3 2	

5. **Re-arrang Array 5.** Cho mảng $A[]$ gồm n số nguyên dương. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng sao cho nguyên tắc số lớn nhất ở đầu, số nhỏ nhất thứ 2, số lớn thứ nhì ở vị trí tiếp theo, số nhỏ thứ nhì ở vị trí kế tiếp Ví dụ với mảng $A[] = \{1, 7, 3, 5, 9\}$ ta được mảng được sắp $A[] = \{9, 1, 7, 3, 5\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq A[i] \leq 10^3$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	7 1 5 2
4	9 1 8 2 7 3
1 5 2 7	
6	
1 3 2 7 9 8	

6. **Re-arrang Array 6.** Cho mảng $A[]$ gồm n số nguyên không âm. Ta gọi phần tử $A[i]$ là hợp lệ nếu $A[i] \neq 0$ ngược lại là không hợp lệ. Nhiệm vụ của bạn là hãy sắp đặt lại các phần tử của mảng theo nguyên tắc nếu số $A[i+1]$ ($i=0, \dots, n-2$) là số hợp lệ và $A[i] = A[i+1]$ thì nhân đôi $A[i]$ và thiết lập $A[i+1] = 0$. Sau khi thay đổi, hãy di chuyển các số không hợp lệ vào cuối mảng. Các số hợp lệ phải bảo toàn quan hệ trước sau. Ví dụ với mảng $A[] = \{2, 2, 0, 4, 0, 8\}$ thực hiện theo nguyên tắc trên ta được kết quả $A[] = \{4, 4, 8, 0, 0, 0\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .

- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^5$; $1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	4 4 8 0 0 0
6	4 2 12 8 0 0 0 0 0 0
2 2 0 4 0 8	
10	
0 2 2 2 0 6 6 0 0 8	

7. **Re-arrang Array 7.** Cho mảng A[] gồm n số nguyên không âm. Hãy sắp đặt lại các phần tử trong mảng sao cho khi nối các số lại với nhau ta được một số lớn nhất. Ví dụ với mảng A[] = {54, 546, 548, 60} thực hiện theo nguyên tắc trên ta được số lớn nhất là 6054854654.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^5$; $1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	6054854654
4	998764543431
54 546 548 60	
8	
1 34 3 98 9 76 45 4	

8. **Re-arrang Array 8.** Cho mảng A[] đã được sắp xếp gồm n số nguyên không có số nguyên nào lặp lại. Hãy sắp đặt lại các phần tử trong mảng theo hình lượn sóng với thứ tự tăng của sóng: $A[0] \geq A[1] \leq A[2] \geq A[3] \leq A[4], \dots : A[0] \leq A[2] \leq A[4] \dots; A[1] \leq A[3] \leq A[5] \dots$. Ví dụ với mảng A[] = {1, 2, 3, 4, 5} ta có thể đưa ra kết quả: thực hiện theo nguyên tắc trên ta được số lớn nhất là A[] = {2, 1, 4, 3, 5}.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng A[]; dòng kế tiếp đưa vào n số A[i] của mảng; các số được viết cách nhau một vài khoảng trống.
- T, n, A[i] thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^6$; $0 \leq A[i] \leq 10^7$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	2 1 4 3 5
5	
1 2 3 4 5	

9. **Re-arrang Array 9.** Cho mảng A[] gồm n số nguyên không âm. Hãy sắp đặt lại các phần tử trong mảng theo nguyên tắc:

- Phần tử đầu tiên được thay bằng tích của nó và phần tử kế tiếp.

- Phần tử cuối cùng được thay bằng tích của nó và sau nó.
- Các phần tử còn lại được thay bằng tích của phần tử sau nó và phần tử trước nó.

Ví dụ với mảng $A[] = \{2, 3, 4, 5, 6\}$ thực hiện theo nguyên tắc trên ta được mảng $A[] = \{6, 8, 15, 24, 30\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên đưa vào n là số phần tử của mảng $A[]$; dòng kế tiếp đưa vào n số $A[i]$ của mảng; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^5$; $1 \leq A[i] \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	6 8 15 24 30
5	63 108 56 72 40 30
2 3 4 5 6	
6	
9 7 12 8 6 5	

10. Re-arrang Array 10. 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 (Decreasing). 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
- $T, \text{Length}(A)$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{Length}(A) \leq 9$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
4	12
I	21
D	321
DD	321654798
DDIDDIID	

11. Re-arrang Array 11. Cho mảng $A[]$ gồm n phần tử gồm các số nguyên dương. Mảng $A[]$ được gọi là Bitonic nếu các phần tử của mảng được chia thành hai phần: phần thứ nhất tăng dần, phần thứ hai giảm dần. Mảng $A[]$ được sắp xếp tăng dần cũng là mảng Bitonic khi xem phần thứ hai là rỗng. Tương tự như vậy, mảng $A[]$ được sắp xếp giảm dần cũng là một bitonic. Hãy tìm độ dài dãy con dài nhất của mảng $A[]$ là một Bitonic. Ví dụ với mảng $A[] = \{1, 11, 2, 10, 4, 5, 2, 1\}$ ta có kết quả là 6 tương ứng với độ dài dãy con $\{1, 2, 10, 4, 2, 1\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số lượng phần tử của mảng N ; phần thứ hai đưa vào n số $A[i]$; các số được viết cách nhau một vài khoảng trống.

- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq n \leq 100; 1 \leq A[i] \leq 200$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
2	5
5	6
1 2 5 3 2	
8	
1 11 2 10 4 5 2 1	

12. **Re-arrang Array 12.** Cho mảng $A[]$ gồm n phần tử gồm các số âm và dương. Hãy tìm giá trị lớn nhất tích các phần tử của tất cả các dãy con liên tục trong mảng $A[]$. Ví dụ với mảng $A[] = \{6, -3, -10, 0, 2\}$ ta có kết quả là 180 tương ứng với tích các phần tử của dãy con $\{6, -3, -10\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất đưa vào số lượng phần tử của mảng N ; phần thứ hai đưa vào n số $A[i]$; các số được viết cách nhau một vài khoảng trống.
- $T, n, A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq n \leq 100; 1 \leq A[i] \leq 200$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
3	180
5	120
6 -3 -10 0 2	288
6	
2 3 4 5 -1 0	
10	
8 -2 -2 0 8 0 -6 -8 -6 -1	

13. **Re-arrang Array 13.** Cho mảng $A1[]$ và $A2[]$ gồm n, m phần tử theo thứ tự. Hãy sắp xếp lại các phần tử trong $A1[]$ theo quan hệ thứ tự trong $A2[]$. Phần tử xuất hiện trước trong $A2[]$ và có mặt trước trong $A1[]$ đứng trước; các phần tử xuất hiện trong $A1[]$ nhưng không xuất hiện trong $A2[]$ đứng sau theo thứ tự tăng dần. Ví dụ với mảng $A1[] = \{2, 1, 2, 5, 7, 1, 9, 3, 6, 8, 8\}$, $A2[] = \{2, 1, 8, 3\}$ sau khi sắp xếp ta được $A1[] = \{2, 2, 1, 1, 8, 8, 3, 5, 6, 7, 9\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T .
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm ba dòng: dòng thứ nhất đưa là hai số n, m ; dòng thứ hai đưa vào n số của mảng $A1[i]$; dòng thứ ba đưa vào m số của mảng $A2[i]$; các số được viết cách nhau một vài khoảng trống.
- $T, n, m, A1[i], A2[j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100; 1 \leq n, m \leq 10^6; 1 \leq A1[i], A2[i] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	2 2 1 1 8 8 3 5 6 7 9
11 4	
2 1 2 5 7 1 9 3 6 8 8	
2 1 8 3	

14. **Re-arrang Array 14.** Cho k mảng mỗi mảng gồm n phần tử đã được sắp xếp. Hãy đưa ra kết quả là một dãy đã được sắp xếp. Ví dụ với $k = 3, n = 4$ và mảng

$$A[] = \{ \{1, 3, 5, 7\},$$

{2, 4, 6, 8},
{0, 9, 10, 11}

};

sẽ cho ta kết quả $A[] = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai phần: phần thứ nhất dòng thứ nhất đưa là hai số k, n; dòng thứ hai đưa vào $n \times k$ số của mảng $A[n][k]$ các số được viết cách nhau một vài khoảng trống.
- T, n, k, $A[i][j]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq n \leq 10^3$; $1 \leq k \leq 10$; $1 \leq A[i][j] \leq 10^6$.

Output:

- Đưa ra kết quả mỗi test theo từng dòng.

Input:	Output:
1	1 2 3 4 5 6 7 8 9
3	
1 2 3 4 5 6 7 8 9	

15. Re-arrang Array 15. Cho mảng $A[]$ gồm n số nguyên dương đã được sắp xếp. Hãy sắp đặt lại các phần tử trong mảng theo nguyên tắc lớn nhất, bé nhất, lớn thứ nhì, bé thứ nhì,... Ví dụ với $A[] = \{54, 546, 548, 60\}$ ta nhận được số lớn nhất là 6054854654. Đối với mảng $A[] = \{1, 2, 3, 4, 5, 6\}$ ta nhận được $\{6, 1, 5, 2, 4, 3\}$.

Input:

- Dòng đầu tiên đưa vào số lượng test T.
- Những dòng kế tiếp đưa vào T bộ test. Mỗi bộ test gồm hai dòng: dòng đầu tiên ghi lại số các phần tử của mảng $A[]$. Dòng tiếp theo đưa vào n số phân biệt các phần tử của $A[]$.
- Các số T, n, $A[i]$ thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 10^7$; $1 \leq A[i] \leq 10^7$.

Output:

- Đưa ra theo từng dòng kết quả mỗi test .

Input	Output
2	6 1 5 2 4 3
6	110 10 100 20 90 30 80 40 70 50 60
1 2 3 4 5 6	
11	
10 20 30 40 50 60 70 80 90 100 110	

16. Re-arrang Array 16. Cho số tự nhiên N. Bạn chỉ được phép sử dụng nhiều nhất một phép đổi chỗ để nhận được số lớn nhất nhỏ hơn N. Ví dụ với số $N=12435$, sử dụng một phép đổi chỗ ta nhận được số lớn nhất nhỏ hơn N là 12354. Với số $N=12345$ ta không có phép đổi chỗ.

Input:

- Dòng đầu tiên đưa vào T là số lượng bộ test.
- T dòng tiếp đưa các bộ test. Mỗi bộ test được viết trên một dòng là một xâu ký tự số không có ký tự '0' đầu tiên.
- Các số T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq \text{length}(N) \leq 10^5$;

Output:

- Đưa ra kết quả mỗi test theo từng dòng, trong đó -1 được xem là test không có phép đổi chỗ.

Input	Output
2	12435
12435	-1

12345	
-------	--

17. Re-arrang Array 17. Số đối xứng lớn nhất. Cho số nguyên dương gồm N chữ số. Bạn chỉ được phép thực hiện hai thao tác:

Thao tác A: loại bỏ tất cả các chữ số giống nhau.

Thao tác B: sắp đặt lại vị trí các chữ số.

Hãy tìm số nguyên đối xứng lớn nhất có thể được tạo ra bằng cách thực hiện hai thao tác A hoặc B ở trên. Ví dụ với số $N = 1122233300000998$, ta có thể tạo ra số đối xứng lớn nhất là 910000019 bằng cách thực hiện các thao tác A, B như sau:

Thao tác A: loại bỏ các chữ số 2 ta nhận được số 1133300000998.

Thao tác A: loại bỏ các chữ số 3 ta nhận được số 1100000998.

Thao tác A: loại bỏ các chữ số 8 ta nhận được số 110000099.

Thao tác B: sắp đặt lại các số còn lại 110000099 để được số 910000019.

Input:

- Dòng đầu tiên đưa vào số lượng bộ test T.
- Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test là một số nguyên dương có N chữ số.
- T, N thỏa mãn ràng buộc: $1 \leq T \leq 100$; $1 \leq N \leq 1000$.

Output:

- Đưa ra số nguyên lớn nhất được tạo ra bởi thao tác A, B của mỗi test theo từng dòng.

Input:	Output:
2 12345000 11233300000998	5 910000019