**SEARCHING PROBLEMS**

1. **Linear Search**. Cho mảng A[] gồm n phần tử. Hãy tìm vị trí của phần tử có giá trị X trong mảng A[]. Nếu không tìm thấy X hãy đưa ra -1.

**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 n, X là số các phần tử của mảng A[] và số X cần tìm; dòng tiếp theo đưa vào n số A[i] (1≤i≤n) các số được viết cách nhau một vài khoảng trống.
* T, n, A, X thỏa mãn ràng buộc: 1≤T≤100; 1≤N, X, A[i] ≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2  5 16 9 7 2 16 4 7 98 1 22 57 47 34 18 66 | 4 -1 |

1. **Binary Search**. Cho mảng A[] gồm n phần tử đã được sắp xếp. Hãy đưa ra 1 nếu X có mặt trong mảng A[], ngược lại đưa ra -1.

**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 n, X là số các phần tử của mảng A[] và số X cần tìm; dòng tiếp theo đưa vào n số A[i] (1≤i≤n) các số được viết cách nhau một vài khoảng trống.
* T, n, A, X thỏa mãn ràng buộc: 1≤T≤100; 1≤N, X, A[i] ≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2  5 16 2 4 7 9 16  7 98 1 18 22 34 47 57 66 | 1 -1 |

1. **Missing Number**. Cho mảng A[] gồm n-1 phần tử bao gồm các khác nhau từ 1, 2, .., n. Không có phần từ nào của mảng trùng nhau. Hãy tìm số không có mặt trong mảng A[].

**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 n l; dòng tiếp theo đưa vào n-1 số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, n, A thỏa mãn ràng buộc: 1≤T≤100; 1≤N, A[i] ≤107.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 5 1 2 3 5 10 1 2 3 4 5 6 7 8 10 | 4 9 |

1. **Search in Sorted & Rotated Array**. Một mảng được sắp được chia thành hai đoạn tăng dần được gọi là mảng sắp xếp vòng. Ví dụ mảng A[] = { 5, 6, 7, 8, 9, 10, 1, 2, 3, 4} là mảng sắp xếp vòng. Cho mảng A[] gồm n phần tử, hãy tìm vị trí của phần tử x trong mảng A[] với thời gian log(n).

**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 n và x; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, n, A[i], x thỏa mãn ràng buộc: 1≤T≤100; 1≤N, x, A[i] ≤107.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 10 3 5 6 7 8 9 10 1 2 3 4 10 3 1 2 3 4 5 6 7 8 9 10 | 8 2 |

1. **First & Second Smallest**. Cho mảng A[] gồm n phần tử, hãy đưa ra số nhỏ nhất và số nhỏ thứ hai của mảng. Nếu không có số nhỏ thứ hai, hãy đưa ra -1.

**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 n là số phần tử của mảng A[]; dòng tiếp theo 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≤T≤100; 1≤N, A[i] ≤107.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 10  5 6 7 8 9 10 1 2 3 4 5  1 1 1 1 1 | 1 2 -1 |

1. **Sum Closest Zezo**. Cho mảng A[] gồm n phần tử, hãy tìm cặp phần tử có tổng gần nhất so với 0.

**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 n là số phần tử của mảng A[]; dòng tiếp theo 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≤T≤100; 2≤N ≤103, -106≤A[i] ≤106.

**Output**:

* Đưa ra tổng gần nhất với 0 của cặp phần tử.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 3 -8 -66 -60   6 -21 -67 -37 -18 4 -65 | -68 -14 |

1. **K Lagest Element**. Cho mảng A[] gồm n phần tử, hãy tìm k phần tử lớn nhất của mảng. Các phần tử được đưa ra theo thứ tự giảm dần.

**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 N và K; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, N, K, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤K < N ≤103, 1≤A[i] ≤106.

**Output**:

* Đưa ra tổng gần nhất với 0 của cặp phần tử.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 5 3  10 7 9 12 6  6 2  9 7 12 8 6 5 | 12 10 8  12 9 |

1. **Count number of occurrences**. Cho mảng A[] gồm n phần tử đã được sắp xếp. Hãy tìm số lần xuất hiện số X trong mảng. Nếu số lần xuất hiện số x trong mảng là 0 hãy đưa ra -1.

**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 N và X; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, N, X, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤ N ≤103, 1≤A[i], X ≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 7 2  1 1 2 2 2 2 3  7 4  1 1 2 2 2 2 3 | 4  -1 |

1. **K Lagest Element**. Cho mảng A[] gồm n phần tử đã được sắp xếp. Hãy tìm số lần xuất hiện số X trong mảng. Nếu số lần xuất hiện số x trong mảng là 0 hãy đưa ra -1.

**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 N và X; dòng tiếp theo đưa vào n số A[i]; các số được viết cách nhau một vài khoảng trống.
* T, N, X, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤ N ≤103, 1≤A[i], X ≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 7 2  1 1 2 2 2 2 3  7 4  1 1 2 2 2 2 3 | 4  -1 |

1. **Sum of Primes**. Số tự nhiên N. Hãy tìm cặp số nguyên tố đầu tiên có tổng là N. Nếu không tồn tại cặp số nguyên tố có tổng bằng N, hãy đưa ra -1.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm là một số N được ghi trên một dòng.
* T, N thỏa mãn ràng buộc: 1≤T≤100; 1≤ N ≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 4  8 | 2 2  3 5 |

1. **Tìm X trong A[]**. Cho mảng A[] gồm N phần tử được chia thành hai phần: phần thứ nhất bao gồm các số được sắp theo thứ tự tăng dần; phần còn lại được sắp theo thứ tự giảm dần. Nhiệm vụ của bạn là tìm số X có thuộc mảng A[] hay không với thời gian O(log(n)). Nếu tìm thấy X trong mảng A[] hãy đưa ra vị trí của X trong mảng A, ngược lại đưa ra -1.

**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 là cặp số N, X; dòng tiếp theo là N số A[i] là các phần tử của mảng A[].
* T, N, X, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤ N ≤107, -107≤ X, A[i] ≤107.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 10 4  1 2 3 5 10 9 8 7 6 4  10 4  1 2 3 5 10 9 8 7 6 -1 | 9  -1 |

1. **Cặp có hiệu là X**. Cho mảng A[] gồm N phần tử và số X. Nhiệm vụ của bạn là tìm cặp phần tử A[i] - A[j] = X. Nếu tồn tại A[i] - A[j] = X đưa ra 1, ngược lại đưa ra -1.

**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 là cặp số N, X; dòng tiếp theo là N số A[i] là các phần tử của mảng A[].
* T, N, X, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤ N ≤105, 1≤ X, A[i] ≤105.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 6 78 5 20 3 2 5 80 5 45 90 70 20 80 50 | 1  -1 |

1. **Min Rotated Array**. Mảng A[] được gọi là mảng sắp xếp vòng nếu A[] được chia thành hai đoạn đã được sắp xếp. Đọc từ phần tử nhỏ nhất đến phần tử cuối cùng và vòng lại các phần tử ở phần còn lại ta sẽ được một dãy đã được sắp xếp. Ví dụ dãy A[] = {5, 6, 1, 2, 3, 4} là mảng sắp xếp vòng. Nhiệm vụ của bạn là tìm phần tử nhỏ nhất của mảng sắp xếp vòng với thời gian O(Log(n)).

**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 tiếp theo là N số A[i] là các phần tử của mảng A[].
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤ N ≤105, 1≤ A[i] ≤105.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 5 4 5 1 2 3 6 10 20 30 40 50 5 7 | 1  5 |

1. **First Element Repeating**. Cho mảng A[] gồm N phần tử. Hãy tìm phần tử lặp lại đầu tiên của mảng. Ví dụ với mảng A[] = {5, 6, 1, 2, 1, 4} thì ta có 1 là phần tử đầu tiên lặp lại trong mảng.

**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 tiếp theo là N số A[i] là các phần tử của mảng A[].
* T, N, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤ N ≤106, 1≤ A[i] ≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 2 5 4 5 1 2 1 6 10 20 30 30 20 5 7 | 1  20 |

1. **K Closest to X**. Cho mảng A[] gồm N phần tử đã được sắp xếp. Hãy tìm K phần tử gần nhất của X. Ví dụ với mảng A[] = {1, 3, 5, 7, 9, 11}, X = 7, K=2 thì ta có 2 phần tử gần nhất của 7 là 5 và 9. Chú ý: X có thể có mặt hoặc không có mặt trong mảng A[].

**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, K, X; dòng tiếp theo là N số A[i] là các phần tử của mảng A[].
* T, N, K, X, A[i] thỏa mãn ràng buộc: 1≤T≤100; 1≤ N, K, X ≤106, 1≤ A[i] ≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 1 5 11 23 24 75 89 2 24 | 23 75 |

1. **Common Elements**. Cho mảng A[], B[], C[] gồm N1, N2, N3 phần tử đã được sắp xếp. Hãy đưa ra các phần tử có mặt trong cả ba mảng theo thứ tự tăng dần.

**Input**:

* Dòng đầu tiên đưa vào số lượng bộ test T.
* Những dòng kế tiếp đưa vào các bộ test. Mỗi bộ test gồm bốn dòng: dòng thứ nhất đưa vào N1, N2, N3 là số phần tử của mảng A[], B[], C[]; các dòng tiếp theo đưa vào N1 số A[i], N2 số B[j], N3 số C[k].
* T, N1, N2, N3, A[i], B[j], C[k] thỏa mãn ràng buộc: 1≤T≤100; 1≤ N1, N2, N3 ≤106, 0≤ A[i], B[j], C[k] ≤1018.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 1 6 5 8 1 5 10 20 40 80 6 7 20 80 100 3 4 15 20 30 70 80 120 | 20 80 |

1. **Kth Smallest Element**. Cho mảng A[]gồm N phần tử. Hãy đưa ra các phần tử nhỏ nhất thứ k của mảng.

**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 N,k là số phần tử của mảng A[] và số k; các dòng tiếp theo đưa vào N số A[i].
* T, N, A[i], k thỏa mãn ràng buộc: 1≤T≤100; 1≤ k< N ≤106, 0≤ A[i] ≤106.

**Output**:

* Đưa ra kết quả mỗi test theo từng dòng.

|  |  |
| --- | --- |
| Input: | Output: |
| 1 6 3  10 5 15 50 40 80 | 15 |