Monuments

The famous Registan Square in Samarkand contains N monuments standing in a line that passes through the center of the square. The monuments are indexed from 0 to $N-1$. Monument i (for $0 \leq i < N$) is initially located at integer coordinate $X[i]$. The center of the square is at coordinate 0.

The architects require perfect symmetry with respect to the center. They want to move some (possibly zero) of the monuments so that the final configuration is symmetric about the coordinate 0. Formally:

- Every monument must be placed at integer coordinates.
- Each integer coordinate may contain **zero, one, or multiple monuments**.
- For every integer $x > 0$, the number of monuments at coordinate x must be equal to the number of monuments at coordinate $-x$. Any number of monuments (possibly zero) can be located at coordinate 0.

However, M monuments are ancient and too fragile to be moved. Their indices are $P[0], P[1], \ldots, P[M-1]$. These M monuments must stay at their original coordinates. The remaining $N - M$ monuments can be moved to any integer coordinates. The monuments do not interfere with each other in any way during movements.

The cost of moving a single monument from coordinate a to coordinate b is the absolute difference $|a - b|$. The total cost is the sum of the costs of all the monuments that are moved.

Your task is to find the minimum total cost to make the monument positions symmetric about coordinate 0, or determine that it is impossible to achieve such a symmetry under the given constraints.

Implementation Details

You should implement the following procedure:

```
long long get_cost(std::vector<int> X, std::vector<int> P)
```

- X: a non-decreasing array of length N describing the initial coordinates of the monuments.
- P: a strictly increasing array of length M describing the indices of the ancient monuments.
- This procedure is called exactly once for each test case.

The procedure should return an integer: the minimum total cost to make the positions symmetric, or -1 if it is impossible.

Constraints

- $1 \leq N \leq 500\,000$
- $0 \leq M \leq N$
- $-10^9 \leq X[0] \leq X[1] \leq \ldots \leq X[N-1] \leq 10^9$
- $0 \leq P[0] < P[1] < \ldots < P[M-1] < N$

Subtasks

Subtask	Score	Additional Constraints
1	3	$M = N$
2	4	$M = 0$
3	5	$X[P[j]] < 0$ for each j such that $0 \leq j < M$.
4	6	$N \leq 10$
5	5	$N \leq 19$
6	5	$N \leq 32$
7	13	$N \leq 200$
8	17	$N \leq 4000$
9	13	$M \leq 4000$
10	29	No additional constraints.

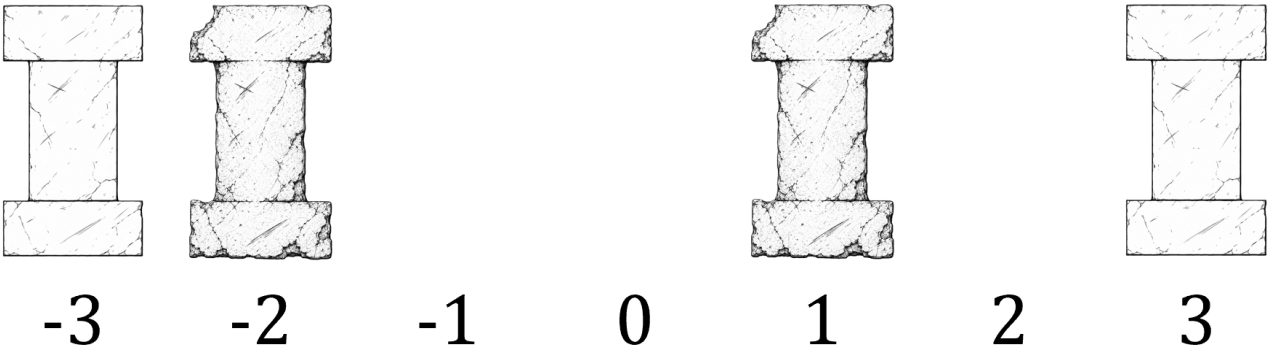
Examples

Example 1

Consider the following call:

```
get_cost([-3, -2, 1, 3], [1, 2])
```

The initial configuration of the monuments is shown in the following figure.

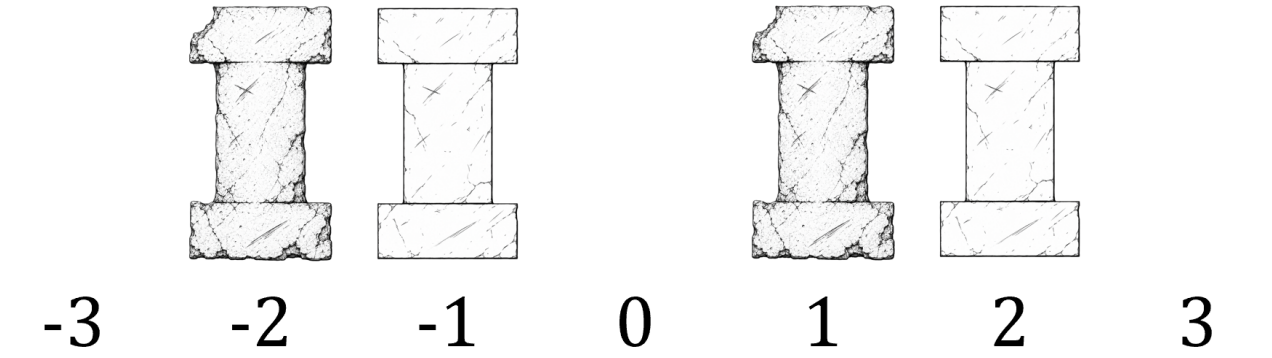


There are $N = 4$ monuments, initially located at coordinates $[-3, -2, 1, 3]$. The indices of the ancient monuments are $P[0] = 1$ and $P[1] = 2$. That is, the monuments at coordinates $X[P[0]] = -2$ and $X[P[1]] = 1$ cannot be moved. The remaining monuments at coordinates -3 and 3 may be moved.

To achieve symmetry with minimum total cost, we should move the monuments as follows:

- Move the monument from coordinate -3 to -1, at a cost of $|(-3) - (-1)| = 2$.
- Move the monument from coordinate 3 to 2, at a cost of $|3 - 2| = 1$.

After these moves, the configuration becomes symmetric.



The total movement cost is $2 + 1 = 3$, so the procedure should return 3.

Example 2

Consider the following call:

```
get_cost([2, 2, 2, 3], [])
```

Here, $M = 0$, which means no monuments are ancient, and all monuments can be moved. One solution with minimum total cost is to move all monuments to coordinate 0. The cost for each monument is:

- $|2 - 0| = 2$ for monuments 0, 1, and 2.
- $|3 - 0| = 3$ for monument 3.

The total cost is $2 + 2 + 2 + 3 = 9$. Hence, the procedure should return 9. Note that there are other solutions with the same total cost.

Example 3

Consider the following call:

```
get_cost([1, 2, 3, 4], [0, 1, 2, 3])
```

All 4 monuments are ancient and cannot be moved. Since their coordinates are all positive, it is impossible to make the configuration symmetric about 0. Hence, the procedure should return -1.

Sample Grader

Input format:

```
N M
X[0] X[1] ... X[N-1]
P[0] P[1] ... P[M-1]
```

Note that the third line may be empty if M is 0.

Output format:

```
C
```

Here, C is the value returned by `get_cost`.