Dynamo

In your house, there is a special machine called the Dinamo. That powers your house.

One day, it breaks. You search the Facegram Marketplace, and manage to find one expert who claims he can fix it.

The expert begins repairing the Dinamo from the first floor. Meanwhile, you are on the third floor, observing how many light bulbs are working. The light bulbs are arranged in a row of N bulbs, each represented as either 0 (off) or 1 (on).

During the repair process, the expert performs two types of operations:

He can flip the state of all bulbs in a given range $[l, r]$, turning $1 \to 0$ and $0 \to 1$.

He may shout: "How many from $[l, r]$?" In this case, he wants you to report the sum of values over all subarrays in that range.

In other words, if you are given a binary array segment $A_l, A_{l+1}, \ldots, A_r$, your task is to compute:

$$\sum_{i=l}^{r} \sum_{j=i}^{r} \sum_{k=i}^{j} A_k$$

Implementation details

You should implement the following procedure.

For C++:

```
void init(int N, int Q, std::string A);
void update(int L, int R);
long long query(int L, int R);
```

For Python:

```
def init(N: int, Q: int, A: str) -> None
def update(L: int, R: int) -> None
def query(L: int, R: int) -> int
```

- `init`: Will be called once.
- `update`: Will be called Q_u times, check constraints and subtasks.
- `query`: Will be called Q_q times, check constraints and subtasks.

Examples

Example 1

Example of calls:

```
init(8, 8, "11101101")
query(1, 1) = 1
update(1, 1)
query(1, 8) = 78
update(2, 7)
query(4, 7) = 8
update(3, 5)
query(4, 6) = 4
query(7, 8) = 4
```

Constraints

- $1 \leq N, Q \leq 3 \cdot 10^5$
- $0 \leq A_i \leq 1$
- $Q_u + Q_q = Q$

Subtasks

1. $N, Q \leq 50$ (10 points)
2. $N, Q \leq 500$ (20 points)
3. $N, Q \leq 5000$ (30 points)
4. No additional constraints (40 points)

Sample grader

line 1: two integers N, Q

line 2: A binary string A of size N

lines 3 to $Q + 2$:

- Each line contains: T, L, R
- If $T = 1$, call $update(L, R)$
- If $T = 2$, call $query(L, R)$