Poprawne dwunawiasowanie **(dwunawiasowanie)**

Limit pamięci: 32 MB Limit czasu: 1.00 s

Poprawnym dwunawiasowaniem nazwiemy każdy ciąg złożony ze znaków `([)]`, który można otrzymać z rekurencyjnej zależności:

- Ciąg pusty jest poprawnym dwunawiasowaniem
- Jeśli `P` jest poprawnym dwunawiasowaniem, to `[P]`, oraz `(P)` są poprawnymi dwunawiasowaniami
- Jeśli `P` i `Q` są poprawnymi dwunawiasowaniami, to `PQ` jest poprawnym dwunawiasowaniem

Przykładowo, `[()[]]` jest poprawnym dwunawiasowaniem, ale `)(]`, lub `[(]` nie są.

Napisz program, który wczyta ciąg nawiasów oraz zapytania o poprawność dwunawiasowania wybranych spójnych podsłów ciągu, wyznaczy dla każdego zapytania czy dwunawiasowanie jest poprawne i wypisze wyniki na standardowe wyjście.

Wejście

W pierwszym wierszu wejścia znajdują się jedna liczba N, oznaczająca długość ciągu.
W drugim wierszu wejścia znajduje się ciąg nawiasów o długości N.
W trzecim wierszu wejścia znajdują się jedna liczba Q, oznaczająca ilość zapytań.
W kolejnych Q wierszach znajduje się opis kolejnych zapytań, po jednym w wierszu. Opis każdego zapytania składa się z dwóch liczb naturalnych L_i, R_i, oddzielonych pojedynczym odstępem. Określają one zapytanie o poprawność zadanego spójnego podciągu dwunawiasowania od L_i-tego znaku do R_i-tego znaku włącznie.

Wyjście

Twój program powinien wypisać na wyjście dokładnie Q wierszy. W i-tym wierszu powinna się znaleźć odpowiedź dla i-tego zapytania. Odpowiedź dla każdego zapytania to jedno słowo `TAK`, jeśli dwunawiasowanie jest poprawne lub `NIE` w przeciwnym przypadku.

Ograniczenia

$1 \leq N, Q \leq 1\ 000\ 000$, $1 \leq L_i \leq R_i \leq N$.
W testach wartych 30% wszystkich punktów zachodzi $1 \leq N, Q \leq 2\ 000$.
W testach wartych 20% wszystkich punktów ciąg składa się tylko ze znaków `()`.

Przykład

Wejście	**Wyjście**
5 [()][5 1 4 2 3 1 2 1 3 1 5	TAK TAK NIE NIE NIE

Wejście	**Wyjście**
4 [(]) 1 1 4	NIE