Problem A. Poruszające się kropki

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Na osi OX znajduje się N kropek, i-ta kropka ma współrzędną x_i. Kropka może być przesunięta w dowolnym kierunku. Kropki są tak małe, że jeśli się pokryją, połączą się w jedną kropkę. Kiedy dwie kropki łączą się, kropka o mniejszym indeksie pokrywa kropkę o większym indeksie, zatem numer identyfikator powstałej kropki jest równy mniejszemu z identyfikatorów kropek.

W jednej sekundzie możesz wykonać następującą operację:

1. Wybierz dowolną kropkę spośród wszystkich obecnie istniejących kropek. Oznaczmy jej indeks jako j.
2. Przesuń kropkę o indeksie j z pozycji x_j na pozycję $(x_j - 1)$ lub $(x_j + 1)$.

Innymi słowy, w jednej sekundzie możesz wybrać kropkę i przesunąć ją w lewo lub w prawo o 1.

Twoim zadaniem jest odpowiedzieć na q zapytań. Dla każdego zapytania oblicz najmniejszą liczbę kropek, które mogą pozostać na płaszczyźnie po upływie t sekund.

Input

Pierwsza linia wejścia zawiera pojedynczą liczbę całkowitą n $(1 \le n \le 2 \cdot 10^5)$ — liczbę kropek.

Druga linia wejścia zawiera n liczb całkowitych x_i $(-10^6 \le x_i \le 10^6)$ — współrzędne kropek na osi.

Następna linia wejścia zawiera jedną liczbę całkowitą q $(1 \le q \le 2 \cdot 10^5)$ — liczbę zapytań.

Następna linia wejścia zawiera q liczb całkowitych t_i $(0 \le t_i \le 10^9)$.

Output

Wypisz q linii — odpowiedzi na odpowiadające zapytania.

Scoring

1. (11 punktów): $n \le 3$;
2. (16 punktów): $0 \le x_i \le 20$;
3. (14 punktów): $t_i \le 20$;
4. (17 punktów): $n, q \le 3000$;
5. (18 punktów): $q \le 100$;
6. (24 punktów): brak dodatkowych ograniczeń.

Examples

standard input	standard output
4 1 2 5 6 2 5 3	1 2
5 8 1 5 11 25 10 0 1 2 3 4 5 6 7 8 9	5 5 5 4 4 4 3 3 3 3

Problem B. Permutacje Mex

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Początkowo mamy tablicę a z n liczbami całkowitymi. Wiadomo, że wszystkie jej elementy są w przedziale $[0; n-1]$ i są parami różne. Innymi słowy, a jest permutacją.

Niech $mex(S)$ oznacza najmniejszą nieujemną liczbę całkowitą, która nie należy do zbioru S. Na przykład, $mex(\{1, 2, 3\}) = 0$, $mex(\{0, 1, 3\}) = 2$, $mex(\{0, 1, 2\}) = 3$.

Tablica F jest budowana w taki sposób, że $F_i = mex(a_1, a_2, \ldots, a_{i-1}, a_i)$. Niech $f(a) = F$.

Budowanie tablicy F na podstawie tablicy a jest dość łatwe, ale niestety tablica a została utracona. Chcesz ją odtworzyć, używając tablicy F, ale okazało się, że istnieje zbyt wiele pasujących tablic. Chcesz wiedzieć, ile istnieje tablic a takich, że $f(a) = F$ modulo 998244353.

Input

W pierwszym wierszu wejścia znajduje się jedna liczba całkowita n ($1 \le n \le 10^6$) — długość permutacji.

W drugim wierszu wejścia znajduje się n liczb całkowitych F_i ($0 \le F_i \le n + 1$) — elementy tablicy F.

Output

Wypisz jedną liczbę całkowitą — liczbę tablic a, które spełniają $f(a) = F$ modulo 998244353.

Scoring

1. (3 punkty): $F_i = n + 1$;
2. (3 punkty): $F_i < F_{i+1}$ dla wszystkich $i < n$;
3. (5 punktów): $n \le 3$;
4. (16 punktów): $n \le 7$;
5. (18 punktów): $n \le 18$;
6. (27 punktów): $n \le 1000$;
7. (28 punktów): brak dodatkowych ograniczeń.

Examples

standard input	standard output
4 1 2 3 4	1
3 1 1 1	0
6 0 0 0 1 1 6	24

Problem C. Nudny problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2.5 seconds
Memory limit:	256 megabytes

Dane jest słowo s długości n. Zdefiniujmy funkcję dla ciągu:

$$f(s) = \sum_{i=1}^{n} \sum_{j=i}^{n} |s_i - s_j|$$

Gdzie $|a-b|$ oznacza odległość między znakami a i b w alfabecie. Innymi słowy, $f(s)$ oznacza sumę $|s_i - s_j|$ dla wszystkich $1 \le i \le j \le n$. Następnie niech

$$cost(s) = \sum_{i=1}^{n} \sum_{j=i}^{n} f(s[i..j])$$

Gdzie $s[i..j]$ oznacza podciąg ciągu s od indeksu i do j. Innymi słowy, $cost(s)$ oznacza sumę $f(s[l..r])$ dla wszystkich $1 \le l \le r \le n$.

Twoim zadaniem jest rozpatrzeć q zapytań. Każde zapytanie zmienia element na pozycji p na c, tj. $s_p = c$. Wypisz $cost(s)$ `mod` 1000000007 przed rozpatrzeniem zapytań oraz po każdym rozpatrzonym zapytaniu.

Input

Pierwsza linia wejścia zawiera pojedynczą liczbę całkowitą n ($1 \le n \le 2 \cdot 10^5$) — długość słowa s.

Druga linia wejścia zawiera słowo s długości n.

Trzecia linia wejścia zawiera pojedynczą liczbę całkowitą q ($1 \le q \le 2 \cdot 10^5$) — liczbę zapytań.

Następne q linii zawierają liczbę całkowitą p ($1 \le p \le n$) i znak c ($c \in \{a, b, \ldots, y, z\}$) — opis zapytań.

Output

Wypisz $q + 1$ liczb całkowitych w osobnych liniach — odpowiedzi na zapytania.

Scoring

1. (3 punkty): $n \le 3$;
2. (5 punktów): $n, q \le 50$;
3. (6 punktów): $n, q \le 100$;
4. (7 punktów): $c, s_i \in \{a, b\}$;
5. (20 punktów): $n, q \le 5000$;
6. (21 punktów): $n, q \le 20000$;
7. (13 punktów): $c, s_i \le$ “n“;
8. (10 punktów): $n, q \le 10^5$;
9. (15 punktów): brak dodatkowych ograniczeń.

Examples

standard input	standard output
3 aba 4 2 a 2 b 3 b 1 c	4 0 4 3 3
17 yuliiaalisadaryna 10 8 l 1 z 3 t 4 x 1 w 3 q 8 o 2 v 9 e 4 h	35140 35140 35276 36788 39884 39516 39132 39668 39824 40112 37072

Problem D. Graf? Jesteś pewien?

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

Jeśli nie wierzysz, nie osiągniesz.

Jeff Long

Początkowo istnieje n wierzchołków i brak krawędzi. Każda dodana krawędź ma napisaną na niej liczbę całkowitą c_i. Definiujemy prostą ścieżkę między dwoma wierzchołkami a i b, należącymi do tej samej składowej, jako najkrótszą ścieżkę od a do b. Rozważmy prostą ścieżkę między dwoma wierzchołkami jako *dobrą*, jeśli każda wartość na tej ścieżce ma parzystą liczbę wystąpień. Musisz odpowiedzieć na q zapytań o 4 typach:

- $1\,u\,v\,c$ $(1 \le c \le 4 \cdot 10^9)$ — dodaj krawędź łączącą wierzchołki u i v z wartością c na niej
- $2\,u\,v$ — powiedz, czy prosta ścieżka między wierzchołkami u i v jest dobra. Jeśli nie ma między nimi ścieżki, wydrukuj -1
- $3\,u$ — powiedz, ile jest par wierzchołków a i b $(1 \le a < b \le n)$, takich że należą do tej samej składowej z u i ścieżka między nimi jest uważana za dobrą
- 4 — powiedz, ile jest par wierzchołków a i b $(1 \le a < b \le n)$, takich że ścieżka między nimi jest uważana za dobrą, a istnieje ścieżka od a do b.

Zapewnione jest, że we wszystkich zapytaniach pierwszego typu krawędź łączy dwie różne składowe.

Input

Pierwsza linia wejścia zawiera dwie liczby całkowite n $(1 \le n \le 2 \cdot 10^5)$ i q $(1 \le q \le 2 \cdot 10^5)$ — liczba wierzchołków i zapytań odpowiednio.

Każda z następujących q linii zawiera opis zapytania. Pierwsza liczba całkowita t_i definiuje typ zapytania.

- $t_i = 1$ definiuje zapytanie pierwszego typu i jest następowane przez 3 liczby całkowite u, v $(1 \le u \ne v \le n)$, c $(1 \le c \le 4 \cdot 10^9)$
- $t_i = 2$ definiuje zapytanie drugiego typu i jest następowane przez 2 liczby całkowite u, v $(1 \le u, v \le n)$
- $t_i = 3$ definiuje zapytanie trzeciego typu i jest następowane przez 1 liczbę całkowitą u $(1 \le u \le n)$
- $t_i = 4$ definiuje zapytanie czwartego typu.

Output

Dla każdego zapytania typu 2 wydrukuj

- `-1`, jeśli nie istnieje prosta ścieżka
- `TAK`, jeśli prosta ścieżka jest uważana za *dobrą*
- `NIE` w przeciwnym razie.

Dla każdego zapytania typów 3 lub 4 wydrukuj jedną liczbę całkowitą — odpowiedź na odpowiadające zapytanie.

Musisz odpowiedzieć na zapytania w kolejności, w jakiej się pojawiają w wejściu.

Scoring

1. (5 punktów): dla wszystkich $t_i = 1$ $u_i = 1$;
2. (5 punktów): $n, q \leq 20$;
3. (7 punktów): $n, q \leq 1000$;
4. (3 punkty): $t_i \leq t_{i+1}$, $t_i \leq 2$, $c = 1$;
5. (6 punktów): $t_i \leq t_{i+1}$, $t_i \leq 2$, $c \leq 8$;
6. (11 punktów): $t_i \leq t_{i+1}$, $t_i \leq 2$;
7. (9 punktów): $q = n$, $t_i = 1$ dla $1 \leq i < n$, $t_q = 4$;
8. (17 punktów): istnieje liczba całkowita e taka, że wszystkie $t_i = 1$ dla wszystkich $1 \leq i \leq e$, a dla wszystkich $e < j \leq q$ $t_i \neq 1$;
9. (9 punktów): $c \leq 1000$, graf i testy są generowane losowo, tzn. dla każdego wierzchołka v $(2 \leq v \leq n)$ losowo wybieramy p_v $(1 \leq p_v < v)$, następnie generujemy losową permutację $perm$ i ustawiamy $v = perm_v$, $p_v = perm_{p_v}$. Typ zapytania jest wybierany losowo (nie wybierzemy pierwszego typu, jeśli nie ma krawędzi do dodania). Wszystkie wartości w zapytaniach są wybierane losowo;
10. (28 punktów): brak dodatkowych ograniczeń.

Example

standard input	standard output
5 11	-1
1 1 2 1	1
2 1 3	NO
1 2 3 1	2
3 2	YES
2 1 2	NO
1 2 4 2	2
1 2 5 2	
3 3	
2 4 5	
2 1 4	
4	