Nawiaski

Zdefiniujmy *poprawne wyrażenie nawiasowe* jak następuje:
- () oraz [] są poprawnymi wyrażeniami nawiasowymi;
- jeśli A jest poprawnym wyrażeniem nawiasowym, to (A) oraz [A] również są poprawnymi wyrażeniami nawiasowymi;
- jeśli A oraz B są poprawnymi wyrażeniami nawiasowymi, to ich sklejenie AB również jest poprawnym wyrażeniem nawiasowym;

Jeśli weźmiemy dowolne poprawne wyrażenie nawiasowe, w którym występuje przynajmniej jedna para nawiasów kwadratowych – czyli [i odpowiadający] – i zamienimy w nim każdy nawias kwadratowy (zarówno otwierający, jak i zamykający) na **otwierający** nawias okrągły, otrzymamy wyrażenie, które nazwiemy *zaburzonym wyrażeniem nawiasowym*.

Na przykład, `((` oraz `((((()))` są zaburzonymi wyrażeniami nawiasowymi. Pierwsze powstaje z poprawnego wyrażenia nawiasowego `[]`. Drugie można uzyskać z czterech poprawnych wyrażeń: `[](((())),([](())),(([]()))` lub `((([])))`.

Twoim zadaniem jest, dla danego zaburzonego wyrażenia nawiasowego, obliczyć, z ilu poprawnych wyrażeń nawiasowych mogło ono powstać.

Wejście

Pierwszy wiersz pliku **brackets.in** zawiera jedną parzystą liczbę całkowitą N ($2 \le N \le 30000$) – długość danego zaburzonego wyrażenia nawiasowego. Drugi wiersz zawiera N znaków `(` i `)` – dane wyrażenie.

Wyjście

Jedyny wiersz pliku **brackets.out** powinien zawierać jedną liczbę całkowitą – resztę z dzielenia szukanej liczby wyrażeń przez 1 000 000 009.

Przykłady

Dane wejściowe (plik `brackets.in`)	Dane wyjściowe (plik `brackets.out`)	Odpowiadające poprawne wyrażenia nawiasowe:
4 **((()**	**2**	[](), ([])

Dane wejściowe (plik `brackets.in`)	Dane wyjściowe (plik `brackets.out`)	Odpowiadające poprawne wyrażenia nawiasowe:
8 **((((((((**	**14**	[][][][], [[]][][], [[]][[]], [][][[]], [[[]]][], [[][]][], [][[][]], [][[[]]], [[[[]]]], [[][[]]], [[[]][]], [[][][]], [[[][]]], [[[[]]]]

Ocenianie

W testach wartych 20 punktów zachodzi warunek N≤50
W testach wartych 45 punktów zachodzi warunek N≤1000.