**JAVA CASES**

**Problem**

You are given the sequence of Nucleotides of one strand of DNA through a string S*S* of length N*N*. S*S* contains the character A, T, C, *A*, *T*, *C*, and G*G* only.

knows that:

* A*A* is complementary to T*T*.
* T*T* is complementary to A*A*.
* C*C* is complementary to G*G*.
* G*G* is complementary to C*C*.

Using the string S*S*, determine the sequence of the complementary strand of the DNA.

**Input Format**

* First line will contain T*T*, number of test cases. Then the test cases follow.
* First line of each test case contains an integer N*N* - denoting the length of string S*S*.
* Second line contains N*N* characters denoting the string S*S*.

**Output Format**

For each test case, output the string containing N*N* characters - sequence of nucleotides of the complementary strand.

**Constraints**

* 1≤*T*≤100
* 1≤*N*≤100
* S contains A, T, C, and G only

**Sample 1:**

Input

4

4

ATCG

4

GTCC

5

AAAAA

3

TAC

Output

TAGC

CAGG

TTTTT

ATG

1. **Problem**

There are 1010 problems in a contest. You know that the score of each problem is either 11 or 100100 points.

came to know the total score of a participant and he is wondering how many problems were actually solved by that participant.

Given the total score P*P* of the participant, determine the number of problems solved by the participant. Print -1−1 in case the score is invalid.

**Input Format**

* First line will contain T*T*, number of test cases. Then the test cases follow.
* Each test case contains of a single line containing a single integer P*P* - denoting the number of points scored by the participant.

**Output Format**

For each testcase, output the number of problems solved by the participant or -1−1 if the score is invalid.

**Constraints**

* 1≤*T*≤1000
* 0≤*P*≤1000

**Sample 1:**

Input

5

103

0

6

142

1000

Output

4

0

6

-1

10