**4 тур - 9.11.2015  по  15.11.2015.**

точка входу для відправлення розв'язків

[http://134.249.159.199//cgi-bin/new-client?contest\_id=29](http://134.249.159.199/cgi-bin/new-client?contest_id=29)

**Задача 1. Щедрий Петрик(20 балів)**

**Ім’я вхідного файлу: input.txt**

**Ім’я вхідного файлу: output.txt**

**Ліміт часу: 2с.**

У щедрого Петрика є декілька яблук. В силу своєї природної щедрості, кожен раз, коли він зустрічає якого-небудь свого друга, він дивиться на яблука, які в нього є і віддає другові половину.

Але Петрик не однаково любить всіх своїх друзів, тому деяким з них він віддає половину яблука, а декому половину всіх яблук, які в нього є. При цьому з окоміром у Петрика не так добре, як із щедрістю, і ділити яблука більше ніж на дві частини, у нього не виходить. Тому, якщо він зустрічає друга, а нього неціле число яблук, то він вимушений віддати половину яблука.

Вранці у Петрика було *n* яблук, а за день Петрик зустрів *k* друзів. З’ясуйте, скільки яблук у нього могло залишитися ввечері.

**Вхідні дані.** Вхідний файл містить два цілих числа*n –* кількість яблук у Петрика і *k* - кількість друзів, яких він зустрів за день (1 ≤ *n* ≤ 1000, 1 ≤ k ≤ 1000).

**Вихідні дані.** Перший рядок вихідного файлу повинен містити число *m* – кількість варіантів відповіді на запитання, скільки яблук може бути в Петрика ввечері. Наступний рядок повинен містити *m* дійсних чисел, відсортованих за зростанням – варіанти відповідей.

**Приклад введення і виведення.**

|  |  |
| --- | --- |
| input.txt | output.txt |
| 6 1 | 2  3.0 5.5 |

**Задача 2. Дужкова послідовність (100 балів)**

**Ім’я вхідного файлу: input.txt**

**Ім’я вхідного файлу: output.txt**

**Ліміт часу: 1с.**

Юна програмістка Ада нещодавно дізналася на уроці інформатики про арифметичні вирази. Вона зацікавилася питанням, що трапиться, якщо з арифметичного виразу видалити все, крім дужок. Ввівши запит у своєму улюбленому пошуковику, вона з'ясувала, що математики називають послідовності дужок, які могли б зустрічатися в деякому арифметичному виразі, правильними дужковими послідовностями. Так, послідовність () (()) є правильною дужковою послідовністю, тому що вона може, наприклад, зустрічатися у виразі (2 + 2) :( 3- (5-2) + 4), а послідовності (() і () ) (не є такими. Легко бачити, що існує п'ять правильних дужкових послідовностей, що складаються рівно з шести дужок (по три дужки кожного типу - відкриваючих і закриваючих):

((())), (() ()), (() ) (), () (()) і () () ().

Ада зацікавилася найпростішими перетвореннями правильних дужкових послідовностей. Для початку Ада вирішила обмежитися додаванням дужок у послідовність. Вона дуже швидко з'ясувала, що після додавання однієї дужки послідовність перестає бути правильною, а ось додавання двох дужок іноді зберігає властивість правильності. Наприклад, при додаванні двох дужок у різні місця послідовності () () можна отримати послідовності (() ()), (()) (), () (()) і () () (). Легко бачити, що при будь-якому способі додавання двох дужок із збереженням властивості правильності одна з нових дужок повинна бути відкриваючою, а інша - закриваюча. Ада хоче підрахувати кількість різних способів додавання двох дужок у задану правильну дужкову послідовність так, щоб знову вийшла правильна дужкова послідовність. На жаль, з'ясувалося, що ця кількість може бути в деяких випадках дуже великою. Ада розрізняє способи отримання послідовності по позиціях доданих дужок в отриманій послідовності. Наприклад, навіть при додаванні дужок у найпростішу послідовність () можна отримати іншу правильну дужкову послідовність сімома способами: **()** (), **(**(**)**), **(**()**)**, (**()**), (**(**)**)**, () **()**, (**) (**). Тут додані дужки виділені жирним шрифтом.

Таким чином, якщо в отриманій послідовності додана відкриваюча дужка стоїть в позиції *i*, а додана закриваюча - в позиції *j*, то два способи, відповідні парам *(i1, j1)* і *(i2, j2)*, вважаються різними, якщо *i1≠ i2* або *j1≠ j2*. Потрібно написати програму, яка за заданою правильною дужковою послідовностю визначає кількість різних описаних вище способів додавання двох дужок.

**Вхідні дані.** Вхідний файл складається з одного не порожнього рядка, що містить рівно *2n* символів: *n* відкриваючих і *n* закриваючих круглих дужок. Гарантується, що цей рядок є правильною дужковою послідовністю.

**Вихідні дані.** Виведіть у вихідний файл кількість різних способів додавання в задану послідовність двох дужок таким чином, щоб вийшла інша правильна дужкова послідовність.

**Приклад введення і виведення.**

|  |  |
| --- | --- |
| input.txt | output.txt |
| () | 7 |
| ()() | 17 |
| (()) | 21 |