Problem C. Coin Game

Input file:	`coins.in`
Output file:	`coins.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

After lunch, Vasya's and Petya's pockets commonly become full of coins received as change. Therefore they enjoy the following game.

They put coins in a row in some order and then make moves in turn. As a move, one can:

- take a few coins in a row starting with the leftmost;
- take a few coins in a row starting with the rightmost;
- choose an integer d ($d > 1$), leave each d-th coin (that is, coins at positions d, $2 \cdot d$, $3 \cdot d$, ..., counting from the leftmost coin) and take the rest.

The game ends when there are no coins left. The goal of each player is to maximize the amount of money taken.

Games played according to these rules used to pass too quickly, so the players invented an additional restriction: in a move, you cannot take more than some appointed amount of money in total.

Vasya and Petya laid out all the coins. Help them to determine how many each of them gets if both players play optimally, and Vasya starts the game.

Note that Vasya and Petya use modern Russian coins. We would like to remind you that in Russia, coins in denominations of 1, 5, 10, 50, 100 (1 ruble), 200 (2 rubles), 500 (5 rubles) and 1000 kopeks (10 rubles) are in circulation.

Input

Input contains several test cases. The number T ($1 \le T \le 20$) of test cases is given in the first line of input. Then the test cases follow.

Each test case is described by two lines. The first one contains two integers N and M ($1 \le N \le 250$, $1000 \le M \le 10\,000$): the number of coins put in the row and the limit of money per move in kopeks. The second line contains N integers a_1, a_2, ..., a_N where each is one of $\{1, 5, 10, 50, 100, 200, 500, 1000\}$.

The sum of values of N over all test cases is not greater than 1000.

Output

For each test case, output a single line with two numbers: the amounts of kopeks Vasya and Petya win if they both play optimally.

Example

coins.in	coins.out
3 5 1000 1 1 1 1 1 8 1000 1 1000 1 1000 10 1000 5 1000 5 1000 1 1000 5 1000 10	5 0 3000 1017 1016 1000