ECOO '16 R1 P1 - Pass or Fail

Points:	5	**Problem type**	Implementation
Time limit:	30.0s	**Allowed languages**	All
Memory limit:	64M		

In Ms. Echo's ICS4U class there are 4 components that determine a student's final grade: Tests, Assignments, Projects and Quizzes. She changes the weights on each of these components from year to year. Last year it was 20% tests, 20% assignments, 50% projects and 10% quizzes, but who knows what it will be this year? To pass the course, a student has to get 50% or more on the weighted average of all four components.

For example, last year Rosa got 98% on the tests, 85% on assignments, 76% on projects and 100% on the quizzes. That means her mark is:

$$98 \times 20\% + 85 \times 20\% + 76 \times 50\% + 100 \times 10\% = 19.6 + 17 + 38 + 10 = 84.6$$

Ms. Echo plays hardball $-$ she never passes a student with less than 50%, even if that student got 49.9999%. All the marks are in for this year. How many students will be passing?

The input will contain 10 testcases.

The first line of each test case contains four integers W_T, W_A, W_P and W_Q separated by spaces, representing the weights of the four components $(0 \le W_T, W_A, W_P, W_Q \le 100$ and $W_T + W_A + W_P + W_Q = 100)$.

This is followed by a line with a single integer N representing the number of students in the class ($1 \le N \le 35$). The next N lines each contain four integers T_i, A_i, P_i and Q_i, separated by spaces, representing the marks of an individual student (out of 100) for each component $(1 \le i \le N$ and $0 \le T_i, A_i, P_i, Q_i \le 100$). Your program should output a single integer for each test case representing the number of students who passed the course that year.

Note that the sample data below contains only 4 test cases but the test data will contain 10.

Sample Input

```
72 4 8 16
7
68 89 4 93
79 5 74 49
38 89 62 41
24 96 49 56
73 32 17 55
65 37 64 73
8 99 94 80
4 85 0 11
2
```

```
57 84 70 57
81 1 85 31
88 1 3 8
6
60 76 21 84
61 86 1 61
54 49 41 78
6 38 74 83
66 39 68 72
82 16 19 16
92 8 0 0
4
66 3 93 84
14 32 68 17
72 59 43 1
47 53 69 89
```

Sample Output

```
4
1
5
2
```