B Top 2000

Problem

When the year 2000 was approaching, the Dutch radio station Radio 2 launched the Top 2000: a chart containing the 2000 most popular singles of all times. The chart was based on votes, cast by the listeners of Radio 2. It was broadcasted in the last week of 1999.

Because of the enormous success of the Top 2000, the radio station decided to compile and broadcast a new list every year. This implied in particular that every year, the Radio 2 program makers had to solve a large and complicated scheduling problem. The first few years, they did this by hand. Since the problem came back every year, they decided it would be worth your while hiring a smart student to write a computer program for it.

So what is the problem? Once the votes of the listeners have been cast, the 2000 singles in this year's Top 2000 are known, and so is the order in which they have to be played: from number 2000 (the least popular) to number 1 (the most popular). Each single has a known length, so one can easily calculate how much time one would need to play all 2000 singles, *provided that one can broadcast continually for a sufficiently long time.*

Unfortunately, every hour, there is a five-minute break for commercials and the latest news. Hence, the 2000 singles must be scheduled over blocks of 55 minutes. No single is played (partly or as a whole) more than once. In particular, no single is split over different blocks.

It is unlikely that the singles (with their known lengths) that have to be played in one block, require exactly 55 minutes. If together they are longer, then they cannot be played completely. The only way to deal with this, is to cut some parts of the singles. If, on the other hand, together they are shorter than 55 minutes, the remaining time must be filled with (spoken) information by a DJ.

Neither of these 'solutions' is really appreciated, because the listeners of Radio 2 like to hear complete songs, and do not like to hear the (usually boring) talk of a DJ. In particular, there is a penalty for every minute of a song that is cut, and there is a penalty for every minute that is filled with the talk of a DJ. The task is to schedule the 2000 singles over blocks of 55 minutes, such that the total penalty is minimal.

The number of blocks is not fixed. However, it must be an integer number, since it is common practice that a radio program fills a number of *complete* blocks. For simplicity, we assume that the length of a single is an (integer) number of minutes. At least one second of every single must be played.

Finally, if the Radio 2 manager is happy with the performance of the computer program for the Top 2000, he considers using (or selling) it for other single charts, as well. Therefore, the number of entries (2000 in the Top 2000) and the block size (55 minutes on Radio 2) must not be fixed, but must be read from the input.

Input

The first line of the input contains a single number: the number of test cases to follow. Each test case has the following format:

- One line with two integers N and M separated by a single space, satisfying $1 \leq N \leq 50,000$ and $15 \leq M \leq 100$: the number of singles to be played, and the number of minutes per block, respectively.
- One line with two integers A and B separated by a single space, satisfying $1 \leq A, B \leq 1,000$: the penalty for every minute of music that is cut, and the penalty for every minute that is filled with (spoken) information by a DJ, respectively.
- One line with N integers, separated by single spaces: the lengths (in minutes) of the N singles in the chart, in the order in which they have to be played. Each integer x on this line satisfies $1 \leq x \leq 20$.

Output

For every test case in the input, the output should contain a single number, on a single line: the minimum (total) penalty of a schedule for the N singles in the input.

Example

Input

```
3
10 25
2 1
8 7 3 5 4 2 9 4 3 4
16 55
4 1
14 9 13 13 6 15 7 8 13 7 5 11 10 11 9 14
15 28
1 2
7 9 7 5 8 7 6 10 5 9 7 9 6 10 5
```

Output

```
4
0
19
```