ACM-ICPC Live Archive

4613 Mountain Road

In the Franconian Switzerland, there is a narrow mountain road. With only a single lane, this is a bottleneck for two-way traffic. Your job is to schedule incoming cars at both ends so that the last car leaves the road as early as possible. Each car is specified by three values: the direction in which it is going, the arrival time at the corresponding beginning of the road, and the driving time this car needs to get through, provided it is not slowed down by other cars in front. Cars cannot overtake each other on the mountain road, and reordering cars in the queues at the ends of the road is not allowed. For safety reasons, two successive cars going in the same direction may not pass any point of the road within less than 10 seconds. This ensures that the second car will not crash into the first car if the latter brakes hard. However, if another car passes in the other direction in between, it will be clear that the road is empty, so in this case, this rule does not apply.

Input

The first line of the input consists of a single integer c ($1 \leq c \leq 200$), the number of test cases. Then follow the test cases, each beginning with a single line consisting of an integer n ($1 \leq n \leq 200$), the number of cars you are to consider in this test case. The remainder of each test case consists of n lines, one line per car, starting with a single upper case letter ('`A`' or '`B`'), giving the direction in which the car is going. Then follow, on the same line, two integers t ($0 \leq t \leq 100000$) and d ($1 \leq d \leq 100000$), giving the arrival time at the beginning of the road and the minimum travel time, respectively, both in seconds. Within a test case, the cars are given in order of increasing arrival time, and no two cars will arrive at the same time.

Output

For each test case, print a single line consisting of the point in time (in seconds) the last car leaves the road when the cars are scheduled optimally.

Sample Input

```
2
4
A 0 60
B 19 10
B 80 20
A 85 100
4
A 0 100
B 50 100
A 100 1
A 170 100
```

Sample Output

```
200
270
```

1213 Sum of Different Primes

A positive integer may be expressed as a sum of different prime numbers (primes), in one way or another. Given two positive integers n and k, you should count the number of ways to express n as a sum of k different primes. Here, two ways are considered to be the same if they sum up the same set of the primes. For example, 8 can be expressed as $3 + 5$ and $5 + 3$ but they are not distinguished.

When n and k are 24 and 3 respectively, the answer is two because there are two sets $\{2, 3, 19\}$ and $\{2, 5, 17\}$ whose sums are equal to 24. There are no other sets of three primes that sum up to 24. For $n = 24$ and $k = 2$, the answer is three, because there are three sets $\{5, 19\}$, $\{7, 17\}$ and $\{11, 13\}$. For $n = 2$ and $k = 1$, the answer is one, because there is only one set $\{2\}$ whose sum is 2. For $n = 1$ and $k = 1$, the answer is zero. As 1 is not a prime, you shouldn't count $\{1\}$. For $n = 4$ and $k = 2$, the answer is zero, because there are no sets of two different primes whose sums are 4.

Your job is to write a program that reports the number of such ways for the given n and k.

Input

The input is a sequence of datasets followed by a line containing two zeros separated by a space. A dataset is a line containing two positive integers n and k separated by a space. You may assume that $n \leq 1120$ and $k \leq 14$.

Output

The output should be composed of lines, each corresponding to an input dataset. An output line should contain one non-negative integer indicating the number of ways for n and k specified in the corresponding dataset. You may assume that it is less than 2^{31}.

Sample Input

```
24 3
24 2
2 1
1 1
4 2
18 3
17 1
17 3
17 4
100 5
1000 10
1120 14
0 0
```

Sample Output

```
2
3
1
0
```

```
0
2
1
0
1
55
200102899
2079324314
```