

DWITE '11 R3 #3 - Combo Discounts

Time limit: 2.0s **Memory limit:** 64M

DWITE, December 2011, Problem 3

A generic fast-food chain had been cutting costs by hiring increasingly cheaper and less trained staff, so now they have a problem — the cashiers can no longer be trusted to correctly apply combo discounts when customers do *want fries with that*. Management figures it would be the easiest to just let computers figure out if there's an applicable discount that could be applied to the order.

The input will contain 5 test cases. Each case will begin with integer $0 \leq N \leq 10$, the number of deals available, followed by N lines in the form of `<discount> <number of items> <item1> <item2> ...`. All of the prices are in cents, so \$1.42 discount would be just an integer: 142. All items are described by a single word, separated by spaces. There will be at least one item, and no more than five. This quantity is described by second item — number of items.

E.g.: `142 3 burger fries drink`

This is followed by integer $1 \leq M \leq 10$, the number of orders in current test case, followed by M lines, each in form of:
`<number of items> <item1> <item2> ...`

The output will print the largest sum of discounts applicable to each order. A discount is applicable if every item listed on the discount description also appears in the order, and every such item had not already contributed to another discount. If there are conflicting applicable discounts (e.g. `burger1 drink`, `burger2 drink`, and the order is `burger1 burger2 drink`), pick the configuration that will produce the largest sum.

Sample's explanation:

The first case doesn't have any discounts, so it's 0 for every line.

The second case has two discounts available. 2-1: only `burger1 fries` is applicable, so discount 100. 2-2: both `burger1 fries` and `burger2 fries` are possible, but there is only one set of `fries`, so we should pick the larger discount; that would be 150. 2-3: Same as previous, but now there are two sets of `fries`, so we could separate the order into two separate discounts; $100 + 150 = 250$. **Ordered items don't necessarily appear in the same order as the discount specification.**

Sample Input

```
0
1
2 burger fries
2
100 2 burger1 fries
150 2 burger2 fries
3
3 burger1 fries fries
3 burger1 burger2 fries
4 burger1 burger2 fries fries
```

Sample Output

```
0
100
150
250
```

Problem Resource: [DWITE](#)