Haitang and Uma Musume

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In this problem, you need to simulate the simplified calculation of training attributes in the game **Uma Musume Pretty Derby**.

Note that the rules in this problem are different from the original game.

Specifically, you will receive the information of an **Uma Musume** and 6 **Support Cards**, as well as several **trainings**, and you need to output the values of each attribute of the **Uma Musume**.

All **Uma Musume** have 6 abilities, which are **speed** *speed*, **stamina** *sta*, **power** *power*, **guts** *guts*, **wisdom** *wis*, and **skill points** *skill*.

An **Uma Musume** has 10 attributes, which are the initial values $speed_0, sta_0, power_0, guts_0, wis_0$ of the first 5 abilities and their training bonuses $speed_\times, sta_\times, power_\times, guts_\times, wis_\times$.

A **Support Card** has 13 attributes, including the friendship bonus $friend$, the motivation effect enhancement $drive_+$, the training effect enhancement $train$, the initial value bonus of the first 5 attributes $speed'_0, sta'_0, power'_0, guts'_0, wis'_0$, and the training bonus of the first 5 attributes $speed_+, sta_+, power_+, guts_+, wis_+$.

In a single game, you will choose an **Uma Musume** and 6 **Support Cards**. Then, you will let your **Uma Musume** do several **training**.

Each training consists of 6 attributes, including the training camp status *summer*, the overweight status *weight*, the current motivation *drive*, the type of training *type*, the level of training lv, and the set of support cards present (including whether each support card is in a friendship training state) S. lv will not be given directly, but the calculation method is as follows:

- During the training camp, lv is fixed at 5.
- Outside the training camp, lv starts at 1.
- After completing 4 times of the same type of training **outside** the camp, the lv of the **specific type** of training will increase by 1, but not exceeding 5.

In the following text, $\sum_{all}$ represents all carried support cards, $\sum_{present}$ represents all support cards present in the current training, and $\prod_{friends}$ represents all support cards present in the current training and in a friendship training state.

The initial number of skill points for the **Uma Musume** is 120, and the initial values of the remaining abilities are $\min(1200, X_0 + \sum_{all} X'_0)$, where X represents one among the 6 ability types.

The increase in an **Uma Musume**'s attribute after a training can be calculated according to the following formula, and we will provide a detailed explanation for each item:

$$\Delta X = \lfloor (base_{lv,type,X} + \sum_{present} X_+) \cdot (\prod_{friends}(1 + 0.01 \cdot friend)) \cdot (1 + 0.01 \cdot \sum_{present} train) \cdot (1 + coef_{drive} \cdot (1 + 0.01 \cdot \sum_{present} drive_+)) \cdot (1 + 0.01 \cdot X_\times) \cdot (1 + \sum_{present} 0.05) \rfloor$$

- Base training value $(base_{lv,type,X} + \sum_{present} X_+)$

The $base_{lv,type,X}$ array will be provided in the appendix. Specifically, this represents the base value for the increase of each ability, for each type of training, at each level.

$\sum_{present} X_+$ represents the sum of the attribute bonuses of all support cards present, and $skill_+$ is always equal to zero.

- Friendship bonus ($\prod_{friends}(1 + 0.01 \cdot friend)$)

For each **Support Card** with a **friendship training**, we add one to its friendship bonus and then multiply them all together.

- Training effect enhancement ($1 + 0.01 \cdot \sum_{present} train$)

We sum up the training bonuses of all **Support Cards** present and add one.

- Motivation bonus ($1 + coef_{drive} \cdot (1 + 0.01 \cdot \sum_{present} drive_{+})$)

The motivation ranges from $[0, 4]$, and $coef_i$ are respectively $-0.2, -0.1, 0, 0.1, 0.2$.

For each **Support Card** present, we sum up their motivation bonuses and add one, then multiply by the motivation coefficient.

- **Uma Musume** growth rate ($1 + 0.01 \cdot X_{\times}$)

This parameter is only related to the **Uma Musume**'s attributes, and $skill_{\times}$ is always equal to zero.

- Number of support cards present ($1 + \sum_{present} 0.05$)

Each present **Support Card** will provide a 0.05 bonus.

- Other rules

In particular, if $weight = 1$, the $\Delta speed$ will be forced to become 0, i.e. the speed cannot be increased.

After training, if the first 5 abilities exceed 1200, we will take the minimum of 1200, note that this rule does not apply to skill points.

You need to print the 6 abilities of the **Uma Musume** after each training.

Input

The first line contains 10 integers $speed_0, sta_0, power_0, guts_0, wis_0, speed_{\times}, sta_{\times}, power_{\times}, guts_{\times}, wis_{\times}$ ($50 \le speed_0, sta_0, power_0, guts_0, wis_0 \le 200$, $0 \le speed_{\times}, sta_{\times}, power_{\times}, guts_{\times}, wis_{\times} \le 30$), representing the **Uma Musume**.

Each of the next 6 lines contains 13 integers
$friend, drive_{+}, train, speed'_0, sta'_0, power'_0, guts'_0, wis'_0, speed_{+}, sta_{+}, power_{+}, guts_{+}, wis_{+}$
($0 \le friend, drive_{+}, train, speed'_0, sta'_0, power'_0, guts'_0, wis'_0, speed_{+}, sta_{+}, power_{+}, guts_{+}, wis_{+} \le 30$), representing a **Support Card**.

The next line contains an integer n ($1 \le n \le 72$) — the number of **trainings**.

Each of the next n lines begins with 5 integers $summer, weight, drive, type, |S|$ ($0 \le summer, weight \le 1$, $0 \le drive, type \le 4$, $0 \le |S| \le 5$), representing a **training**. Then $|S|$ pairs of integers x_i, y_i ($1 \le x \le 6$, $0 \le y \le 1$) follow, representing the x_i-th **Support Card** is present in this training, and $y = 1$ means it has the **friendship training**.

It is guaranteed that in each training, x_i are pairwise different.

Output

For each test case, print one line contains 6 integers — the abilities of the **Uma Musume**.

Example

standard input	standard output
71 117 70 102 90 0 10 0 20 0 25 30 15 0 0 0 0 0 0 1 0 0 0 30 30 15 0 0 0 0 30 0 0 0 0 1 25 0 15 0 0 0 0 0 0 0 0 0 2 30 30 10 0 0 20 0 0 0 0 1 0 0 30 30 0 0 0 0 20 0 1 0 1 0 0 30 0 10 0 0 0 0 0 0 0 0 1 0 9 0 1 0 0 1 5 0 0 0 1 0 4 1 1 2 0 3 0 4 0 0 1 2 0 3 1 0 3 0 6 0 0 0 3 0 5 1 1 2 1 3 1 4 1 5 1 0 1 4 0 3 2 1 4 0 6 1 1 0 3 1 0 1 1 4 2 1 3 1 1 0 3 3 3 4 1 5 0 6 1 1 1 4 4 1 4 0	71 117 94 122 120 121 89 119 105 122 125 124 89 120 113 123 128 127 178 128 169 123 152 143 178 128 189 127 155 149 178 143 189 132 155 151 178 156 210 132 158 154 194 156 228 174 158 159 194 156 229 174 176 164

Note

Here is the table of *base*:

Table $base_1$:

$type/X$	$speed$	sta	$power$	$guts$	wis	$skill$
0	10	0	5	0	0	2
1	0	9	0	4	0	2
2	0	5	8	0	0	2
3	4	0	4	8	0	2
4	2	0	0	0	9	4

Table $base_2$:

$type/X$	$speed$	sta	$power$	$guts$	wis	$skill$
0	11	0	5	0	0	2
1	0	10	0	4	0	2
2	0	5	9	0	0	2
3	4	0	4	9	0	2
4	2	0	0	0	10	4

Table $base_3$:

$type/X$	$speed$	sta	$power$	$guts$	wis	$skill$
0	12	0	5	0	0	2
1	0	11	0	4	0	2
2	0	6	10	0	0	2
3	4	0	4	10	0	2
4	3	0	0	0	11	4

Table $base_4$:

$type/X$	$speed$	sta	$power$	$guts$	wis	$skill$
0	13	0	5	0	0	2
1	0	12	0	4	0	2
2	0	6	11	0	0	2
3	4	0	4	11	0	2
4	3	0	0	0	12	4

Table $base_5$:

$type/X$	$speed$	sta	$power$	$guts$	wis	$skill$
0	14	0	5	0	0	2
1	0	13	0	4	0	2
2	0	7	12	0	0	2
3	5	0	5	12	0	2
4	4	0	0	0	13	4

The author set this problem after he had uninstalled the game.