Problem 235: Roll for Initiative

Difficulty: Medium

Author: Brett Reynolds, Bethesda, Maryland, United States

Originally Published: Code Quest 2024

Problem Background

If you've ever played a computer game such as *Fortnite*, *World of Warcraft*, or *Final Fantasy*, then you know that having the best possible gear or the best possible statistics on your character greatly improves your chances of success. Very often, a player can't even attempt very difficult challenges until they're sufficiently "geared up." This sort of statistics-based system is a common feature throughout the entire genre of role-playing games (RPGs), and players put a lot of work into determining the optimal "build" for characters in each game.

However, there isn't simply one perfect setup for these games; typically, games allow players to play in a variety of different ways. For example, someone playing as a soldier may value their physical strength more than their magical skills, whereas a sorcerer would prefer the opposite. Each "class" will have its own priorities regarding which abilities should be favored over which others.

The original RPG, *Dungeons & Dragons*, currently has players establish six "ability scores" for their characters. A common (and humorous) explanation of these scores is:

- Strength is your ability to smash a tomato
- Dexterity is your ability to dodge a tomato thrown at you
- Constitution is your ability to survive eating a poisoned tomato
- Intelligence is knowing that a tomato is a fruit
- Wisdom is knowing that you shouldn't put a tomato in a fruit salad
- Charisma is your ability to sell a tomato-based fruit salad

As with other RPGs, a character will be more successful (and less likely to be killed or suffer some other horrible fate) if they maximize abilities appropriate to their class; other abilities can be largely ignored.

Problem Description

You are working with a gaming company to develop a character generator for *Dungeons & Dragons*. Your team has determined the ideal ability priorities for each class. To assist new players in creating their characters, your program will randomly generate ability scores for them and assign them to the appropriate abilities automatically. Your task is to write the code that handles the score assignments; someone else has already sorted out the random number generation, and will be feeding your code the numbers they generate.

Your program will use the common abbreviations for each of the ability scores:

- Strength = STR
- Dexterity = DEX
- Constitution = CON
- Intelligence = INT
- Wisdom = WIS
- Charisma = CHA

For each character, you must assign the given score values to each of the character's abilities according to the priority given for that character's class. The highest score should be assigned to the ability with the highest priority; the second-highest should be assigned to the ability with the second-highest priority, and so on.

Sample Input

The first line of your program's input, received from the standard input channel, will contain a positive integer representing the number of test cases. Each test case will include the following lines:

- A line containing two positive integers, separated by a space:
    - C, the number of classes from which players may choose
    - P, the number of players generating characters
- C lines containing the following information, separated by spaces:
    - The name of a class, which will contain only uppercase letters
    - Each of the six ability score abbreviations (STR, DEX, CON, INT, WIS, CHA), listed in order from the highest priority to the lowest priority for the identified class.
- P lines containing the following information, separated by spaces:
    - The name of a class, which will contain only uppercase letters and will match one of the classes listed in the previous section
    - Six positive integers, each ranging from 1 to 20 inclusive, representing the randomly generated score values to be assigned to the player's character.

```
1
3 3
WARRIOR STR CON DEX CHA WIS INT
ROGUE DEX CHA INT WIS STR CON
WIZARD INT DEX WIS CON CHA STR
WARRIOR 5 7 9 11 13 15
ROGUE 8 8 10 11 12 13
WIZARD 10 10 13 13 15 15
```

Sample Output

For each test case, your program must print the following:

- A line containing the name of the character's class, in all uppercase letters
- A line containing the text "STR: #", where # is replaced with the value of the character's Strength score

- A line containing the text “DEX: #”, where # is replaced with the value of the character’s Dexterity score
- A line containing the text “CON: #”, where # is replaced with the value of the character’s Constitution score
- A line containing the text “INT: #”, where # is replaced with the value of the character’s Intelligence score
- A line containing the text “WIS: #”, where # is replaced with the value of the character’s Wisdom score
- A line containing the text “CHA: #”, where # is replaced with the value of the character’s Charisma score

```
WARRIOR
STR: 15
DEX: 11
CON: 13
INT: 5
WIS: 7
CHA: 9
ROGUE
STR: 8
DEX: 13
CON: 8
INT: 11
WIS: 10
CHA: 12
WIZARD
STR: 10
DEX: 15
CON: 13
INT: 15
WIS: 13
CHA: 10
```