The Informatics Canteen

Time Limit: 1 s
Memory Limit: 256 MB

Description

After two years of closure due to the pandemic, the canteen at Informatics Department is finally open :D There are several stalls there that offers various foods and cold beverages, starting from snacks like fried fish dumplings with peanut sauce (*batagor*) to a more heavier dish like fried rice with sunny side-up fried egg (*nasgor telor*).

Yesterday, one of the canteen staff asks whether or not you can help getting their work easier by writing a simple accounting program for them. The program can add a new menu, get the information of available menu, storing customer orders, and at the end of the day, view the total earnings of the canteen. You just learn about struct in the past weeks, so you told the staff you're up to it. He smiles happily and said he will help you with some details. Each menu in the canteen consists of $name$ and price, and each menu will be contained in their own shelves with a number X, ranged from 0 to 30. The menu name may contain space character so you might want to use `"%[^\n]"` inside your `scanf()` function. More details are provided in *Input Format*

Input Format

There are several commands for the programs, each with their own action as specified below. Also don't forget to check *Example* for a more direct example

- `ADD` to add a new menu. The next three lines after this command is the X (the shelf number), $name$, and $price$
- `INFO X` to get the information of a menu in the shelf number X
- `ORDER X N`, where X is the menu in the shelf number X to order and N is how much of the menu the customer ordered.

- `CLOSE` to close the canteen. It is guaranteed that this command will always be called at the end of the day

Output Format

If the issued command are other than those listed above, print `Unknown command!`, otherwise, the following is the output format of each command

- When a new menu is added via `ADD`, print `Added menu #X name`, where X is the shelf number and name is the name of the menu. But if the shelf is already filled, print `Menu #X is already exist`
- When getting information via `INFO`, if the menu exists, print `#X name Rp price`, but if not, print `Menu #X doesn't exist`
- After done `ORDER` ing, print `Ordered Nx name for Rp total price`, where N is the number of menu ordered and total price is the $price \times N$, but if the menu doesn't exist, print `Menu #X doesn't exist`
- After the command `CLOSE`, print `Earnings: Rp total`, where total is the sum of all menu price ordered the whole day, then, print `TCanteen is closing.. thank you for the visit!`

Constraints

$0 \leq X \leq 30$
$1 \leq price \leq 10^5$
$1 \leq strlen(name) \leq 2^5$

Example

Input
ADD 0 batagor 6000 ADD 1 nasi goreng telor 12000 ADD 2 nasi goreng ayam krispi 15000 INFO 0 INFO 1 INFO 2 INFO 5

```
ORDER 0 3
ORDER 1 2
CLOSE
```

Output

```
Added menu #0 batagor
Added menu #1 nasi goreng telor
Added menu #2 nasi goreng ayam krispi
#0 batagor Rp 6000
#1 nasi goreng telor Rp 12000
#2 nasi goreng ayam krispi Rp 15000
Menu #5 doesn't exist
Ordered 3x batagor for Rp 18000
Ordered 2x nasi goreng telor for Rp 24000
Earnings: Rp 42000
TCanteen is closing.. thank you for the visit ^-^
```

Explanation

Quite self-explanatory I reckon. I've also added some color highlights to mark which line input corresponds to which output. Happy solving!

Hints

- When you need to compare two strings, use `strcmp()` from `<string.h>`, with the syntax `strcmp(str1, str2)`. If str1 and str2 are the same, the function will return 0, otherwise, it will be non-zero.
  Example: to check whether the issued command is "ADD"

  ```
  if (strcmp(cmd, "ADD") == 0) {
        // statement
  } else if …
  ```
- Handling a string input in C is quite tricky, so if you find out your program skipped some input after receiving a string via `scanf()`, you can try calling `getchar()` before the `scanf()`