

The Informatics Canteen

Time Limit: 1 s

Memory Limit: 256 MB



Updates

EN

- Fixed the *Output Format* of the command `ADD` to match *Example*, sorry there was some mistake before

ID

- Maaf sebelumnya ada sedikit kesalahan dengan *Output Format* dari perintah `ADD` jika dibandingkan dengan *Example*, sudah diperbaiki

Description

EN After two years of closure because of the pandemic, the canteen at Informatics Department is finally open :D There are many 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 telur*).

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 more details later (read more in *Input Format*). The canteen have a really large storage room for storing foods. All menu will be stored in their own shelves in a really large storage room. There are 1.000.000 shelves inside in which each shelf is given a unique id number, from 0 to 1.000.000

ID Setelah dua tahun tutup karena pandemi, kantin di Departemen Teknik Informatika akhirnya dibuka kembali :D Ada banyak stan di sana yang menyediakan

beragam makanan dan juga minuman dingin, mulai dari cemilan seperti batagor, pisang coklat, sampai makanan utama seperti nasi goreng telur, soto ayam, dan masih banyak lagi.

Kemarin, salah seorang staf kantin menyanyimu tentang apakah kamu bisa membantu mempermudah pekerjaan mereka dengan membuat sebuah program akuntansi sederhana. Program ini dapat menambah menu baru, mendapatkan informasi menu yang tersedia, menyimpan pesanan pelanggan, dan melihat total pendapatan ketika kantin hendak tutup. Kamu telah belajar *struct* beberapa minggu lalu, jadi kamu bersedia membantu mereka. Staf itupun tersenyum senang dan berkata dia akan memberi detail lengkapnya setelah ini (baca lebih lanjut di *Input Format*). Semua menu akan disimpan di rak masing-masing di dalam suatu ruang penyimpanan yang sangat luas. Terdapat 1.000.000 rak di dalamnya dimana setiap rak diberi nomor id unik mulai 0 hingga 1.000.000

Input Format

EN There are several commands for the programs, each with their own action as specified below

- **ADD *id*** to add a new menu to the system, where *id* is the unique ID number of the menu. This command will be followed with two lines. The first line contains *name*, the menu name. The second line contains *price* and *stock*, where *price* is the price of the menu and *stock* the number of the available stock
- **REMOVE *id*** to remove a menu from the system, where *id* is the ID number of the menu to be removed
- **INFO *id*** to get the information of a menu based on its *id* number
- **ORDER *id* *amount*** for ordering menu, where *id* is the ID number of the menu to be ordered and its *amount*
- **DISC *id* *percentage*** for giving a menu a special discount, where *id* is the menu to be discounted and *percentage* is a floating point number—how much the discount is
- **CLOSE** for closing the canteen. It is guaranteed that this command will be called at the end of the day

ID Ada beberapa perintah dengan fungsinya masing-masing yang bisa diterima program ini, yaitu

- **ADD *id*** untuk menambah menu baru ke dalam sistem, dimana *id* adalah nomor ID unik dari menu. Perintah ADD akan diikuti 2 baris tambahan. Baris pertama adalah *name*, yaitu nama menu. Baris selanjutnya adalah *price* dan *stock*, dimana *price* adalah harga dari menu dan *stock* adalah banyak stok menu yang tersedia
- **REMOVE *id*** untuk menghapus sebuah menu dari sistem, dimana *id* adalah nomor ID menu yang hendak dihapus
- **INFO *id*** untuk mendapat informasi suatu menu dari nomor *id*-nya
- **ORDER *id* *amount*** untuk memesan menu, dimana *id* adalah nomor ID menu dan *amount* adalah jumlah menu yang dipesan

- `DISC id percentage` untuk memberi diskon pada suatu menu, dimana *id* adalah ID menu yang diberi diskon dan *percentage* adalah bilangan desimal—besar diskon dari menu tersebut
- `CLOSE` untuk menutup kantin. Dijamin bahwa perintah ini akan selalu dipanggil di akhir

Output Format

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

- When a new menu is added via `ADD`, print `Menu added`. But if the menu is already exist on the shelf, print `Menu is already exists`
- When removing a menu via `REMOVE`, print `Menu removed`. But if the menu doesn't exist, print `Menu does not exist`
- When obtaining information via `INFO`, there will be four lines printed, the last line being an empty line. First line is `#id name`, second is `Price : RpXXX`, and last is `Stock : stockx`, where *XXX* is the menu price and *stock* is the number of available stock. If the menu doesn't exist, print `Menu does not exist`
- When ordering a menu, print `Ordered amountx name for RpXXX`, where *amount* is the amount ordered, *name* is the name of the ordered menu, and *XXX* is the total price, but if the *amount* is 0, print `So you want to order or what`. If the stock isn't enough to fulfill the ordered amount, print `Apologize, the stock is not enough`. Else, if the menu doesn't exist, print `Menu does not exist`
- When giving discount to a menu, print `Discount added`. If the menu doesn't exist, print `Menu does not exist`
- After the command `CLOSE`, print `Earnings: RpXXX`, where *XXX* is the total earnings for the whole day. Then, print `Informatics canteen is closing.. thanks for the visit!`

ID Jika perintah yang diterima tidak ada dalam daftar, cetak `Unknown command`. Namun jika ada, berikut adalah keluaran yang harus dicetak untuk tiap baris perintah

- Ketika berhasil menambah menu baru dengan `ADD`, cetak `Menu added`. Tapi jika rak telah terisi, cetak `Menu is already exists`
- Ketika berhasil menghapus menu dengan `REMOVE`, cetak `Menu removed`. Tapi jika menu tidak ditemukan, cetak `Menu does not exist`
- Ketika menemukan informasi dengan perintah `INFO`, ada empat baris keluaran, dimana baris terakhir adalah baris kosong. Baris pertama adalah `#id name`, kedua `Price : RpXXX`, dan ketiga adalah `Stock : stock`, dimana *XXX* adalah harga menu dan *stock* adalah jumlah stok yang tersedia. Jika menu tidak ditemukan, cetak `Menu does not exist`
- Ketika berhasil memesan menu, cetak `Ordered amountx name for RpXXX`,

dimana *amount* adalah jumlah yang dipesan, *name* adalah nama menu, dan *XXX* adalah total harga menu, tetapi jika *amount* adalah 0, cetak `So you want to order or what`. Jika stok tidak bisa mencukupi menu yang dipesan, cetak `Apologize, the stock is not enough`. Hanya jika menu tidak ditemukan, cetak `Menu does not exist`

- Ketika berhasil mendiskon suatu menu, cetak `Discount added`, tetapi jika menu tidak ditemukan, cetak `Menu does not exist`
- Setelah perintah `CLOSE`, cetak `Earnings: RpXXX`, dimana *XXX* adalah total pendapatan kantin selama sehari. Setelah itu, cetak `Informatics canteen is closing.. thanks for the visit!`

Constraints

$$0 \leq id \leq 1000000$$

$$0 \leq price, stock, amount \leq INT_MAX$$

$$0\% \leq discount \leq 100\%$$

$$1 \leq strlen(name) \leq 2^6$$

Examples

Input
<pre> ADD 0 Batagor 6000 100 ADD 1 Nasi goreng telur 11000 50 ADD 2 Nasi soto ayam 13000 40 INFO 0 INFO 1 REMOVE 2 INFO 2 ORDER 0 5 ORDER 1 1 DISC 1 10 ORDER 1 2 CLOSE </pre>
Output
<pre> Menu added Menu added </pre>

```
Menu added
#0 Batagor
Price : Rp6000
Stock : 100x

#1 Nasi goreng telur
Price : Rp11000
Stock : 50x

Menu removed
Menu does not exist
Ordered 5x Batagor for Rp30000
Ordered 1x Nasi goreng telur for Rp11000
Discount added
Ordered 2x Nasi goreng telur for Rp19800
Earnings: Rp60800
Informatics canteen is closing... thanks for the visit!
```

Explanations

EN Quite self-explanatory I reckon. But if you need more clarification, do hit us up using the "clarifications" feature of the DOMJudge. Happy solving!

ID Harusnya udah cukup jelas, tapi jika butuh klarifikasi lebih lanjut, pake aja fitur "clarifications" dari DOMJudge buat sampein ke asisten. Selamat mengerjakan!

Hints

EN You may need to use `malloc()` and `free()`

ID Coba pakai `malloc()` dan `free()`