

رامین و مسئله‌ی ریاضی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۵۰ مگابایت

محمد می‌خواهد توانایی ریاضی رامین را به چالش بکشد. او یک مقدار شروع و پایان را ارائه می‌دهد که بازه‌ای از اعداد صحیح را شامل می‌شود (شامل جمله نقاط آغازی و پایانی). رامین باید تعداد اعداد مربع کامل را در آن بازه مشخص کند.

نکته: عدد مربع کامل یک عدد صحیح است که برابر با مربع یک عدد صحیح می‌باشد.

مثال:

$$l = 24$$

$$r = 49$$

در اینجا ۳ عدد مربع کامل در بازه‌ی $[24, 49]$ وجود دارد (۲۵، ۳۶ و ۴۹). بنابراین پاسخ برابر ۳ است.

از آنجا که رامین حوصله‌ی محاسبه ندارد، از شما می‌خواهد تا به او کمک کنید و به q پرسش آن پاسخ دهید.

ورودی

ورودی شامل دو خط است. در خط اول q آمده که نشان‌دهنده‌ی تعداد پرسش‌هایی است که رامین از شما می‌کند و در هر یک از q خط بعدی، دو عدد آمده است که به ترتیب نشان‌دهنده‌ی l و r است.

$$1 \leq q \leq 100$$

$$l \leq r$$

$$1 \leq l, r \leq 10^9$$

خروجی

خروجی برنامه‌ی شما باید شامل q خط باشد که در خط i ام باید پاسخ مسئله برای i امین پرسش را چاپ کنید.

ورودی نمونه

```
2
3 9
17 49
```

خروجی نمونه

```
2
3
```

در پرسش اول، در بازه‌ی $[3, 9]$ تنها اعداد ۴ و ۹ مربع کامل هستند. و در پرسش دوم در بازه‌ی $[17, 49]$ تنها اعداد ۲۵، ۳۶ و ۴۹ مربع کامل هستند.

ساعت کار

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

یکی از مشتریان ما در فروشگاه زنجیره‌ای خود یک دستگاه شارژ خودرو برقی نصب و راه اندازی کرده است همچنین نیاز دارد تا روشن و یا خاموش بودن این دستگاه را برنامه ریزی کند و دستگاه تنها زمانی به ارائه سرویس بپردازد که روشن باشد.

دستگاه او در سه سطح دسترسی مختلف قابل برنامه ریزی است:

۱. فروشگاه زنجیره‌ای Tenant
۲. دکان Store
۳. دستگاه Station

برای مثال ممکن است که Tenant یک برنامه برای ساعات روشن بودن به دستگاه بدهد و دکانی (Store) که دستگاه در آن قرار دارد یک برنامه دیگر و همچنین می‌توان یک برنامه اختصاصی به خود دستگاه (Station) داد. برنامه‌ها می‌توانند با یکدیگر اشتراک (تداخل) داشته باشند و در صورت تداخل اولویت با سطح دسترسی با شماره‌ی بیشتر است.

در حالت پیش‌فرض دستگاه برنامه‌ی هفتگی دارد که طبق آن روشن و خاموش می‌شود، این برنامه در ورودی به شما داده می‌شود. اما مشتری ما که این برایش کافی نیست ساعاتی را به عنوان استثنا تعریف می‌کند که در آن ساعات دستگاه طبق آن کار کند.

حال او از شما می‌خواهد که پس از دریافت برنامه‌ها به ازای هر یک از q زمان داده شده تشخیص دهید که آیا دستگاه روشن است یا خاموش.

قوانین:

- استثناها به ساعات کاری اولویت دارند.
- استثنایی که برای یک Station تعریف می‌شود بر استثنایی که برای یک Store تعریف می‌شود اولویت دارد.
- همچنین استثنایی که برای یک Store تعریف می‌شود بر استثنایی که برای یک Tenant تعریف می‌شود

اولویت دارد.

- ترتیب اولویت استثناها به صورت زیر است:

- دستگاه Station

- دکان Store

- فروشگاه زنجیره‌ای Tenant

- سطوحی دسترسی سه مورد Tenant, Store, Station هستند.

- در صورتی که چند استثنا با یک سطح دسترسی موجود بود استثنایی که در لیست دستورات بالاتر است اولویت بیشتری دارد.

ورودی

خط اول به ترتیب شامل اعداد n ، m و q است.

$$1 \leq n, m \leq 200$$

$$1 \leq q \leq 10$$

سپس در n سطر بعدی برنامه‌ی هفتگی می‌آید، که در هر خط آن ابتدا روز آن دستور در هفته و سپس دو زمان شروع و پایان دستور می‌آید. برای مثال:

Monday 8:00 12:00

به این معنی است که در روز Monday هر هفته دستگاه از ساعت ۸ تا ۱۲ روشن است.

- تضمین می‌شود ساعات داده شده به صورت $MM : HH$ می‌باشند که در آن:

$$0 \leq HH \leq 23$$

$$0 \leq MM \leq 59$$

- توجه داشته باشید برای مثال ساعت ۸ و ۷ دقیقه به صورت 8:7 می‌آید.
- همچنین روزهای هفته از بین رشته‌های زیر است:

[Saturday, Sunday, Monday, Tuesday, Wednesday, Thursday, Friday]

سپس در m خط بعدی استثناها می‌آیند، که در هر خط آن ابتدا سطح دسترسی استثنا، سپس دو زمان شروع و پایان بازه استثنا و در آخر دستور مورد نظر می‌آیند. برای مثال:

Tenant 2020-05-20T09:00 2020-05-20T11:00 closed

به این معنی است که دستگاه در بازه زمانی مشخص شده خاموش می‌باشد. تضمین می‌شود دستور مورد نظر یکی از closed یا open است.

در هر یک از q خط بعدی نیز یک زمان خواسته شده توسط مشتری آمده است.

خروجی

در q خط خروجی به ازای هر زمان تشخیص دهید آیا دستگاه در این زمان روشن است یا خاموش، اگر روشن بود عبارت true و در غیر این صورت false چاپ کنید.

مثال

ورودی نمونه ۱

```
16 4 2
Monday 8:00 12:00
Monday 13:00 17:30
Tuesday 8:00 12:00
Tuesday 13:00 17:30
Thursday 8:00 12:00
Thursday 13:00 17:30
Wednesday 8:00 13:00
Friday 8:00 12:00
Friday 13:00 20:00
Saturday 10:00 13:00
Monday 6:30 19:00
Tuesday 6:30 19:00
Thursday 6:30 19:00
Wednesday 6:30 14:30
```

Friday 6:30 21:00

Saturday 9:00 14:30

Tenant 2020-05-20T09:00 2020-05-20T11:00 closed

Store 2020-06-01T08:00 2020-06-20T20:00 open

Station 2020-06-01T06:00 2020-06-05T18:00 closed

Station 2020-05-01T00:00 2020-05-02T00:00 closed

2020-05-01T00:00

2020-06-10T00:00

خروجی نمونه ۱

false

true

قرارمون یادت نره!

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

یک شرکت بین‌المللی دارای p کارمند در کشورهای مختلف دنیاست. هر یک از این افراد دارای منطقه زمانی خاص خود و نیز ساعات کاری اختصاصی در منطقه زمانی خود هستند.

کارکنان این شرکت می‌بایست در ساعات کاریشان در m جلسه کاری شرکت کنند. با توجه به ساعات کاری هر فرد در روز داده شده اولین زمان مناسبی که تمامی افراد جلسه بتوانند در آن حضور داشته باشند جلسه برگزار می‌شود برای هر جلسه زمان برگزاری آن را بیابید و آن را بر اساس ناحیه زمانی UTC اعلام کنید. در غیر اینصورت عبارت N/A را برای جلسات غیر قابل برگزاری چاپ کنید.

قوانین برگزاری جلسات:

- یک فرد نمیتواند همزمان در ۲ جلسه حضور داشته باشد.
- یک جلسه در صورتی برگزار می‌شود که همه افراد شرکت کننده در آن جلسه بتوانند به طور کامل در طول ساعت کاری خود در آن حاضر شوند و در غیر اینصورت جلسه لغو می‌شود.
- اولویت جلسات از زیاد به کم است. یعنی تا زمانی که اولین جلسه تعیین تکلیف نشده باشد، هیچ تصمیمی در مورد جلسات بعدی گرفته نمی‌شود. (این لزوماً به این معنی نیست که جلسات به ترتیب زمانی برگزار می‌شوند. ممکن است جلسه اول با توجه به ناحیه زمانی شرکت‌کنندگان آن دیرتر از جلسه دوم برگزار شود. اما در هر صورت تا هر جلسه تعیین تکلیف نشده جلسات بعدی معین نمی‌شوند.)
- یک جلسه باید در زودترین زمان ممکن در بازه زمانی مجاز برای آن جلسه برگزار شود.
- همه‌ی جلسات باید در یک روز بر حسب UTC برگزار شوند.

ورودی

در خط اول ورودی به ترتیب دو عدد p و m آمده که نشان دهنده‌ی تعداد کارمندان شرکت و تعداد جلسات است.

$$1 \leq p, m \leq 100$$

سپس در p خط بعدی برای هر کارمند نام او، اختلاف منطقه زمانی او با UTC و شروع و پایان ساعت کاریش آمده.

تضمین می‌شود نام کارمندان شرکت متمایز است.

- همچنین تضمین می‌شود ساعات داده شده به صورت $MM : HH$ می‌باشند که در آن:

$$0 \leq HH \leq 23$$

$$0 \leq MM \leq 59$$

در $2m$ خط بعدی، برای هر جلسه دو خط ورودی داده شده، در خط اول آن به ترتیب اعداد c و t آمده‌اند که نشان دهنده‌ی تعداد افراد در جلسه و مدت زمان جلسه به دقیقه می‌باشد و در خط بعد از آن c رشته‌ی جدا شده با فاصله آمده که نام افراد شرکت کننده در جلسه است.

برای جزئیات بیشتر به مثال‌ها توجه کنید.

خروجی

برای هر جلسه اگر امکان برگزاری آن وجود داشت زمان شروع جلسه بر اساس ناحیه زمانی UTC و در غیر این صورت عبارت N/A را چاپ کنید.

مثال

ورودی نمونه ۱

```
3 1
Mahdi +01:00 09:00 17:00
Mat +04:30 09:30 17:30
Mamali +01:00 09:30 16:30
2 30
Mamali Mat
```

خروجی نمونه ۱

```
08:30
```


ورودی نمونه ۲

3 3
 Ali +01:00 08:00 17:00
 Jamshid -02:00 07:00 16:00
 James +06:00 07:00 15:00
 2 30
 Ali James
 2 30
 Ali Jamshid
 3 10
 Ali Jamshid James

خروجی نمونه ۲

07:00
 09:00
 N/A

ربات معامله‌گر

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

به تازگی به شرکتی فعال در حوزه بورس پیوستید! تیم معاملات نزد شما می‌آیند و از شما می‌خواهند باری از دوش این تیم بردارید. بعد از مدتی جست‌وجو و جلسات متوجه می‌شوید یکی از بیشترین معاملاتی که انجام می‌دهند، خرید سهم بر اساس قیاس «میانگین متحرک کوتاه و بلند» است.

میانگین متحرک کوتاه که در مقیاس‌های بزرگ معمولاً ۵۰ روزه است، یکی از قابل اطمینان‌ترین نشانگرهای فنی است که معمولاً سرمایه‌گذاران برای تشخیص رشد و پیشرفت سهم به آن رجوع می‌کنند. این میانگین، ضعف یا قدرت یک سهم بسته به قیمت فعلی‌اش را نشان می‌دهد.

میانگین متحرک بلند که در مقیاس‌های بزرگ معمولاً ۲۰۰ روزه است، معمولاً نمایانگر حرکت سهم در درازمدت (۴۰ هفته) است که می‌تواند رو به بالا یا روبه پایین باشد، همچنین از این میانگین می‌توان نقاط دارای حمایت یا مقاومت را تشخیص داد.

شما تصمیم می‌گیرید این موضوع را خودکار کنید. گروه معاملات از شما می‌خواهد این دستورات را دنبال کنید و سپس گزارش عملکرد ربات خرید و فروش خودکار خود را (سود و زیان) اعلام کنید.

قوانین کلی:

۱. وقتی میانگین متحرک کوتاه اکیداً بزرگ‌تر از میانگین متحرک بلند شد، n سهم خریداری کن.
۲. وقتی میانگین متحرک کوتاه اکیداً کمتر از میانگین متحرک بلند شد، n سهم بفروش.
۳. اگر پس از پایان روزی بر روی خرید خود بیشتر از T درصد سود داشتیم، n سهم را بفروش و از بازار خارج بشو.
۴. اگر پس از پایان روزی بر روی خرید خود بیشتر از S درصد ضرر کردیم، n سهم را بفروش و از بازار خارج بشو.
۵. موارد بالا را با دو روش H و $\frac{OHLC}{4}$ انجام و نتیجه را در خروجی چاپ کن.

قوانین جزئی:

- وقتی خرید انجام شد، دیگر سهمی نباید در بازار خریداری شود تا زمانی که فروش انجام گردد.

- وقتی فروش انجام شد، مجدداً خرید آزاد می‌شود.
- ممکن است در اطلاعات دریافتی چند بار عملیات خرید و فروش انجام شود.
- تمام روزها دارای قیمت هستند. هیچ روز تعطیلی وجود ندارد.
- اگر در آخرین روز معاملات (آخرین داده ورودی) هنوز سهم خریداری شده وجود دارد ولی فروش انجام نشده است، باید فروش انجام پذیرد.
- اگر به یکی از دلایل دریافت سود با T درصد یا جلوگیری از ضرر با S درصد فروش انجام شد، دیگر خرید و فروش دیگری تا انتهای مقادیر ورودی نباید انجام بپذیرد.
- قیمت خرید و فروش در هر روز بسته به روش متفاوت است. برای مثال، قیمت خرید و فروش یک روز در حالت H برابر با مقدار H در آن روز و قیمت خرید و فروش در حالت $\frac{OHLC}{4}$ برابر با مقدار $\frac{OHLC}{4}$ در آن روز است.
- میانگین متحرک تنها برای روزهایی محاسبه می‌گردد که تعداد روزهای سپری شده، کمتر از بازه‌ی میانگین متحرک نباشد. برای مثال اگر بازه‌ی میانگین متحرک ۴ روزه باشد، میانگین متحرک برای روزهای ۴ به بعد محاسبه می‌گردد و در ۳ روز نخست، معامله‌ای صورت نمی‌گیرد.

نکته:

میانگین متحرک ساده هر روز از چندین فرمول بدست می‌آید که با یکدیگر تفاوت دارند، اما در این سوال ما از دو روش آن را حساب می‌کنیم:

۱. در روش نخست فقط بالاترین قیمت معاملاتی روز را مدنظر قرار می‌دهیم، این قیمت با نشان H در ورودی قابل دریافت است.
۲. روش دوم، میانگین $\frac{OHLC}{4}$ است. به این ترتیب که برای محاسبه میانگین آن روز، فرمول زیر صدق می‌کند:

$$\frac{O + H + L + C}{4}$$

ورودی

در پنج خط اول ورودی، به ترتیب اعداد n و SMA و LMA و S و T آمده است. در خط ششم عدد d آمده است که نشان‌دهنده‌ی تعداد روزهای پیش‌رو است و در خط ۷ام پس از آن نیز به ترتیب مقادیر O, H, L, C برای روز ۸ام آمده است.

$$1 \leq n \leq 10\,000$$

$$1 \leq SMA, LMA \leq 200$$

$$SMA \leq LMA$$

$$0.1 \leq S, T \leq 100$$

$$1 \leq d \leq 10\,000$$

$$0.1 \leq O, H, L, C \leq 15\,000$$

توضیحات ورودی:

نماد	توضیحات
O	قیمت سهم در زمانی که بازار باز شده است.
H	بالترین قیمت معامله انجام شده در روز
L	پایین ترین قیمت معامله انجام شده در روز
C	قیمت آخرین معامله انجام شده روز
T	میزان درصد سودی که به محض رسیدن به آن باید از بازار خارج شوید
S	میزان درصد ضرری که به محض رسیدن به آن باید از بازار خارج شوید
SMA	تعداد روزهایی است که باید با آن میانگین متحرک کوتاه را محاسبه کنید
LMA	تعداد روزهایی است که باید با آن میانگین متحرک بلند را محاسبه کنید

- نکته: مقادیر $OHLC$ با کاراکتر فاصله از یکدیگر جدا شده‌اند.
- نکته: تمامی ردیف‌ها به ترتیب زمانی از کم به زیاد در اختیار شما قرار خواهند گرفت.

خروجی

در دو خط خروجی، دو روش را چاپ کنید.

در پارامتر اول هر خط، کد خروجی را مطابق جدول زیر وارد کنید و سپس بعد از کاراکتر فاصله مبلغ سود یا زیان را وارد کنید.

جدول کدها:

کد	معنی
A	خرید و فروش به روش $\frac{OHL C}{4}$ تا انتها بدون توقف سود و ضرر انجام شده است.
AT	روبات در روش $\frac{OHL C}{4}$ با دریافت سود متوقف شده است.
AS	روبات در روش $\frac{OHL C}{4}$ با توقف ضرر متوقف شده است.
B	خرید و فروش به روش H تا انتها بدون توقف سود و ضرر انجام شده است.
BT	روبات در روش H با دریافت سود متوقف شده است.
BS	روبات در روش H با توقف ضرر متوقف شده است.

برای نمایش اعداد خروجی، دو عدد اعشاری شامل میزان سود یا ضرر با دو رقم اعشار نشان دهید. برای اعداد صحیح هم از دو رقم 00 برای نمایش اعشار استفاده کنید. همچنین برای نمایش ضرر از عدد منفی استفاده کنید.

ورودی نمونه ۱

```

5
2
4
0.1
2
10
20.0 40.0 10.0 30.0
31.0 52.0 20.0 42.0
11.0 21.0 5.0 19.0
12.0 15.0 11.0 13.0
13.0 27.0 2.0 18.0
12.0 13.0 10.0 13.0
4.0 17.0 2.0 5.0
4.0 15.0 2.0 5.0
5.0 6.0 5.0 6.0
    
```

6.0 8.0 6.0 8.0

خروجی نمونه ۱

AS -25.00

BT 20.00

در مثال داده شده، ربات در الگوریتم‌های گفته شده به صورت زیر عمل می‌کند:

روش $\frac{OHL C}{4}$: در روز ۶ با قیمت $\frac{12+13+10+13}{4}$ می‌خرد و در روز ۷ به علت ضرر بیش از S درصد با قیمت $\frac{4+17+2+5}{4}$ می‌فروشد و از بازار خارج می‌شود.

روش H : در روز ۶ با قیمت ۱۳ می‌خرد و در روز ۷ با قیمت ۱۷ به علت سود بیش از T درصد می‌فروشد.

ورودی نمونه ۲

200

2

4

0.4

5

8

6675.978 6677.458 6675.978 6677.357

6677.397 6678.334 6677.397 6677.735

6676.579 6678.42 6676.579 6677.834

6676.43 6680.03 6676.43 6679.762

6678.962 6680.106 6678.791 6679.898

6676.951 6679.348 6675.723 6678.108

6679.29 6680.403 6679.06 6680.031

6678.836 6679.646 6678.626 6679.377

خروجی نمونه ۲

A 306.60

B 74.60

بهترین قیمت ارز دیجیتال

- محدودیت زمان: ۵ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

ارزهای دیجیتال قیمت‌گذاری متمرکزی ندارند. دارندگان ارز دیجیتال می‌توانند قیمت مورد نظر برای خرید یا فروش را شخصاً در قالب یک سفارش مشخص کنند. مثلاً اگر بخواهید ۲۰ واحد ارزی مفروض به نام *FirstCoin* را به قیمت ۵۰ تومان بخرید، پیغام شما به این شکل خواهد بود (a شناسه‌ی یکتای سفارش خرید شماست که توسط سیستم تولید می‌شود):

```
ADD a buy FirstCoin 50.00 20
```

اگر زمانی قصد فروش مقداری از موجودی ارز دیجیتال خود را داشته باشید، به طور مشابه مقدار و قیمت پیشنهادی خود را به دیگران اعلام می‌کنید. مثلاً برای فروش ۵۰ واحد *FirstCoin* به قسمت ۴۵ تومان، این پیغام را خواهید فرستاد (b شناسه‌ی یکتای سفارش فروش شماست که توسط سیستم تولید می‌شود):

```
ADD b sell FirstCoin 45.00 50
```

موجودی سفارش‌ها ثابت نمی‌ماند و در صورت لغو سفارش یا انجام معامله ممکن است مقداری از مبلغ آن کم شود. مثلاً در صورت فروش ۳۰ واحد از سفارش b پیغام زیر توسط سیستم مخابره خواهد شد:

```
REM b 30
```

مجموعه تمامی پیغام‌های مخابره شده از این دست را اصطلاحاً دفتر سفارش‌ها (Order Book) می‌گویند.

می‌خواهیم برنامه‌ای به نام تعیین قیمت بنویسیم که با دریافت دفتر سفارش‌های چند ارز دیجیتال، «بهترین قیمت» خرید و فروش برای حجم مشخص و ثابتی از هر ارز را در هر لحظه محاسبه کند. بدیهی است که برای خرید مایلیم کمترین پول ممکن را پرداخت کنیم و هنگام فروش بیشترین پول ممکن را دریافت کنیم. معیار ما برای تعیین «بهترین قیمت» نیز همین است. برنامه مورد نظر، عددی را به عنوان حجم هدف (تعداد واحدهای ارز برای خرید یا فروش) دریافت می‌کند و سپس پیغام‌های دفتر سفارش‌ها را خط به خط خوانده و به محض این که تغییری در بهترین قیمت خرید یا فروش هر ارز در حجم مورد نظر ایجاد شود، آن را در خروجی اعلام می‌کند.

توجه کنید که سفارش‌های یک ارز، کاملاً مستقل از سایر ارزها است. بنابراین تبدیل ارز در دفتر سفارش‌ها معنایی

ندارد و قیمت‌های اعلام شده در خروجی برنامه صرفاً بر اساس سفارش‌های ثبت شده برای یک ارز واحد محاسبه می‌شوند.

ورودی

- خط اول از ورودی برنامه، شامل دو عدد است: حجم هدف (Target) و تعداد ارزها (CoinCount) که با یک فاصله از هم جدا شده‌اند:

Target CoinCount

- خط دوم فهرستی از نام ارزهای موجود را نشان می‌دهد که با یک فاصله از هم جدا شده‌اند (نام ارزها، کاراکتر فاصله ندارد):

coin_1 coin_2 ... coin_N

- در خط بعدی عدد n آمده است که نشان‌دهنده‌ی تعداد پیغام‌های موجود در دفتر سفارش است. در n خط بعدی و در هر خط یک پیام آمده است. دو نوع پیام در دفتر سفارش وجود دارد:

۱. ثبت سفارش که قالب پیغام به صورت زیر است:

time ADD id type coin price size

نماد	معنی
time	زمان ثبت سفارش به صورت یک عدد
ADD	یک رشته‌ی حرف ثابت که مشخص کننده پیغام ثبت سفارش است
id	یک رشته‌ی حرفی که شناسه‌ی یکتای سفارش مورد نظر است
type	یک رشته‌ی حرفی که نوع سفارش را مشخص می‌کند و دو مقدار دارد: sell (فروش) و buy (فروش)
coin	نام یک ارز دیجیتال به صورت یک رشته حرفی (بدون فاصله)
price	قیمت پیشنهادی برای هر واحد ارز
size	تعداد واحد ارز موجود در سفارش

۲. تغییر موجودی که قالب پیغام به صورت زیر است:

time REM id size

نماد	معنی
time	زمان ثبت سفارش به صورت یک عدد
REM	یک رشته‌ی حرف ثابت که مشخص کننده پیغام تغییر موجودی است
id	شناسه‌ی یکتای سفارش مورد نظر
size	تعداد واحد کاهش یافته از موجودی سفارش

$$1 \leq Target \leq 200$$

$$1 \leq CoinCount \leq 20$$

$$1 \leq n \leq 100\,000$$

$$1 \leq price \leq 200\,000$$

$$1 \leq size \leq 200$$

خروجی

خروجی برنامه عبارت است از پیغام‌هایی به شکل زیر که خط به خط در خروجی چاپ خواهند شد:

time type coin total_price

نماد	معنی
time	زمان تعیین قیمت که باید مساوی با فیلد time در آخرین پیغام دریافت شده باشد
type	یک رشته‌ی حرف که نوع معامله‌ی مربوطه را مشخص می‌کند و می‌تواند دو مقدار داشته باشد: sell و buy

نام ارز دیجیتال	coin
بهترین قیمت محاسبه شده برای خرید یا فروش	total_price

نکات تکمیلی:

- مقدار *total_price* باید به صورت عددی با دو رقم اعشار گزارش شود. برای مثال به جای 11.4 باید 11.40 نوشته شود.
- در صورتی که سیستم پیشتر قیمتی برای خرید یا فروش یک ارز اعلام کرده باشد، ولی با ویرایش سفارش‌ها، جمع واحدهای موجود در سفارش‌ها به کمتر از *target* برسد (یعنی خرید یا فروش با حجم هدف غیرممکن شود)، فیلد *total_price* با رشته‌ی حرفی NA پر می‌شود.
- سفارشات لزوماً بر اساس زمانشان مرتب نیستند.
- اگر پس از پیغامی در دفتر سفارشات، ارزش ارز مورد نظر تغییری نکند و مقدار قبلی بماند نیازی به خروجی دادن دوباره نیست. (یعنی تنها زمانی خروجی جدید می‌دهیم که مقدار قبلی اعلام شده دیگر معتبر نباشد)

ورودی نمونه ۱

```
200 1
FirstCoin
5
1003 ADD c buy FirstCoin 44.10 100
1008 ADD d buy FirstCoin 44.18 157
1009 ADD e sell FirstCoin 44.38 120
1010 REM d 80
1015 ADD g sell FirstCoin 44.27 100
```

خروجی نمونه ۱

```
1008 sell FirstCoin 8832.56
1010 sell FirstCoin NA
1015 buy FirstCoin 8865.00
```

پس از پیغام اول سفارش جدید برای خرید ثبت شده؛ ولی چون همچنان از حجم هدف کمتر است و تغییری در ارزش حاصل نشده، چیزی در خروجی چاپ نمی‌شود.

در مرحله‌ی بعد ۲۵۷ واحد درخواست خرید ثبت شده است. با توجه به قیمت‌های پیشنهادی، بهتر است ۱۵۷ واحد را از سفارش d (قیمت بیشتر) و ۴۳ واحد از سفارش c انتخاب می‌کنیم که جمعاً ۸۸۳۲.۵۶ تومان بهترین قیمت برای فروش محاسبه می‌شود.

در مرحله‌ی بعد ۸۰ واحد از مجموع سفارش‌های خرید کم می‌شود و کلاً ۱۷۷ واحد باقی می‌ماند. پس فروش ۲۰۰ واحد غیرممکن و مبلغ فروش قبلی نامعتبر است. در مرحله‌ی آخر نیز ۱۰۰ واحد از سفارش g (قیمت کمتر) و ۱۰۰ واحد از سفارش e می‌خریم که جمعاً ۸۸۶۵ تومان پایین‌ترین قیمت فروش محاسبه می‌شود.

ورودی نمونه ۲

```
1 1
FirstCoin
5
1003 ADD c buy FirstCoin 4410 100
1008 ADD d buy FirstCoin 4418 157
1009 ADD e sell FirstCoin 4438 120
1010 REM d 80
1015 ADD g sell FirstCoin 4427 100
```

خروجی نمونه ۲

```
1003 sell FirstCoin 4410.00
1008 sell FirstCoin 4418.00
1009 buy FirstCoin 4438.00
1015 buy FirstCoin 4427.00
```