

## تخصیص سفارش

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

می‌خواهیم سیستم سفارش گرفتن و تخصیص راننده‌ها به سفارش‌ها را برای اسنپ باکس شبیه سازی کنیم.

در ابتدا هیچ راننده و یا سفارشی نداریم. در هر مرحله یا یک راننده به سیستم اضافه می‌شود یا یک سفارش در سیستم ثبت می‌شود یا یک راننده درخواست می‌کند که یک سفارش به او تخصیص شود.

هر راننده بعد از تخصیص یک سفارش مرحله به مرحله گزارش می‌دهد که فرآیند ارسال این سفارش از فرستنده به گیرنده در کدام مرحله قرار دارد.

از شما می‌خواهیم با دریافت این اطلاعات این سامانه را شبیه سازی کنید. همچنین سیستم شبیه ساز شما باید قابلیت پاسخ دادن به درخواست‌هایی که در پایین تر تعریف کردیم را داشته باشد.

### ▼ راننده

راننده ( DRIVER ) سفارش را از فرستنده تحویل گرفته و به گیرنده تحویل می‌دهد. برای هر راننده ویژگی‌های زیر تعریف می‌شود:

نام ( USERNAME ):

یک رشته که فقط شامل حروف کوچک و بزرگ انگلیسی و ارقام است و حداکثر شامل ۲۵ کاراکتر می‌شود.

نوع وسیله ( SERVICE-CATEGORY ):

وسیله‌ی نقلیه راننده را نشان می‌دهد و مقدار آن همواره یکی از موارد زیر است.

- موتور ( BIKE )

- وانت ( VAN )
- کامیون ( TRUCK )

### میزان اعتبار ( CREDIT ):

یک عدد حسابی است که مقدار پولی که راننده در حساب اسنپ خود دارد را نشان می‌دهد. میزان سرمایه اولیه هر راننده به صورت پیشفرض برابر صفر است. میزان اعتبار هر راننده بعد از تحویل یک سفارش (یعنی زمانی که وضعیت سفارش به DELIVERED تغییر کند) به اندازه 80% هزینه سفارش افزایش پیدا می‌کند.

### موقعیت مکانی ( POSITION ):

موقعیت مکانی هر راننده به صورت یک زوج مرتب  $(x, y)$  است که  $x$  و  $y$  دو عدد صحیح هستند. برای نمایش و دریافت یک POSITION ابتدا یک ( سپس عدد  $x$  بلافاصله , سپس با یک فاصله عدد  $y$  و بلافاصله ) نمایش داده می‌شود.

به طور مثال موقعیت مکانی یک راننده می‌تواند به صورت  $(57, 31)$  یا  $(-74, 0)$  باشد اما  $(0, -74)$  یا  $(57, -74)$  نمایش قابل قبولی نیست.

### وضعیت ( STATUS ):

بطور کلی راننده دارای دو وضعیت زیر است:

- آزاد ( FREE )
- مشغول ( BUSY )

اگر یک راننده در حال انجام دادن یک سفارش باشد یعنی در حال رفتن به سمت فرستنده برای دریافت محموله (به عبارت دیگر محموله تخصیص یافته به آن در حالت ARRIVED باشد) یا در حال رفتن به سمت گیرنده برای تحویل محموله (به عبارت دیگر سفارش تخصیص یافته به آن در حالت PICKUP باشد) باشد؛ راننده در وضعیت BUSY و اگر هیچ سفارشی به این راننده تخصیص نشده یا همه

سفارش‌هایی که به او تخصیص شده را تحویل داده است (یعنی وضعیت سفارش‌های تخصیص‌یافته DELIVERED باشد). وضعیت راننده FREE است.

## ▼ سفارش

برای هر سفارش ( ORDER ) ویژگی‌های زیر قابل تعریف است:

### نوع وسیله ( SERVICE-CATEGORY ):

وسیله‌ی مورد نیاز برای جابه‌جایی این سفارش را نشان می‌دهد و مقدار آن همواره یکی از موارد زیر است.

- موتور ( BIKE )
- وانت ( VAN )
- کامیون ( TRUCK )

### موقعیت مکانی فرستنده ( STARTING-POSITION ):

موقعیت مکانی فرستنده به صورت یک زوج مرتب  $(x, y)$  است که  $x$  و  $y$  دو عدد صحیح هستند. نمایش و دریافت موقعیت مکانی فرستنده مشابه موقعیت مکانی راننده است.

### موقعیت مکانی گیرنده: ( FINISHING-POSITION ):

موقعیت مکانی گیرنده به صورت یک زوج مرتب  $(x, y)$  است که  $x$  و  $y$  دو عدد صحیح هستند. نمایش و دریافت موقعیت گیرنده مشابه موقعیت مکانی راننده است.

### هزینه ( COST ):

هزینه هر سفارش با توجه به فرمول زیر محاسبه می‌شود:

فرض کنید  $(x, y)$  موقعیت مکانی فرستنده و  $(x', y')$  موقعیت مکانی گیرنده باشد. اگر تعداد سفارش‌هایی که هم اکنون منتظر راننده با همین نوع وسیله هستند (یعنی وضعیت آن‌ها PENDING است). شامل همین سفارش، برابر  $k$  باشد این هزینه برابر است با:

$$cost = (k + |x - x'| + |y - y'|) \times 100$$

توجه کنید بعد تحویل هر سفارش 80% هزینه به اعتبار ( CREDIT ) راننده اضافه می‌شود و 20% باقی مانده به حساب اسنپ باکس اضافه می‌شود. (با توجه به رابطه بالا این دو عدد همواره عدد صحیح خواهند بود).

شماره سفارش ( ORDER-ID ):

شماره یک سفارش  $n$  است اگر  $n$  امین سفارشی باشد که با موفقیت در سیستم ثبت شده باشد.

وضعیت ( STATUS ):

وضعیت هر سفارش چهار حالت دارد:

- PENDING : هنوز راننده‌ای به این سفارش تخصیص داده نشده است.
- ARRIVED : یک راننده به این سفارش تخصیص داده شده و به سمت فرستنده این سفارش در حال حرکت است.
- PICKUP : راننده این سفارش را از فرستنده تحویل گرفته و در حال حرکت به سمت گیرنده است.
- DELIVERED : سفارش با موفقیت به گیرنده تحویل داده شده است.

منظور از فاصله بین دو نقطه  $(x, y)$  و  $(x', y')$  در این سوال برابر است با:

$$|x - x'| + |y - y'|$$

از شما می‌خواهیم درخواست‌های زیر را برای این سامانه پیاده سازی کنید:

▼ درخواست ADD-DRIVER

این درخواست به معنی اضافه شدن یک راننده ( DRIVER ) به سیستم است.

ADD-DRIVER <DRIVER-USERNAME> <POSITION> <SERVICE-CATEGORY>



به جای <DRIVER-USERNAME> یک رشته که نشان دهنده نام راننده، به جای <POSITION> موقعیت مکانی فعلی راننده که یک زوج مرتب از اعداد صحیح است و به جای <SERVICE-CATEGORY> نوع وسیله راننده که یک رشته برابر BIKE یا VAN یا TRUCK است قرار دارد.

برای مثال یک نمونه از این درخواست به صورت زیر خواهد بود:

```
ADD-DRIVER shahin021 (13, 74) BIKE
```

بعد از دریافت این نوع درخواست، در صورتی که این نام برای راننده‌ای دیگر قبلاً استفاده شده است عبارت user previously added را چاپ کنید و در صورتی که نام راننده قبلاً در سیستم ثبت نشده عبارت user added successfully را چاپ کنید.

میزان اعتبار یک راننده در لحظه اضافه شدن به سیستم برابر صفر است. وضعیت یک راننده بعد از اضافه شدن به سیستم FREE است چون هنوز هیچ سفارشی به او تخصیص نشده است.

#### ▼ درخواست CREATE-ORDER

این درخواست به معنی اضافه شدن یک سفارش (ORDER) به سیستم است.

```
CREATE-ORDER <SERVICE-CATEGORY> <START-POSITION> <FINISH-POSITION>
```



به جای <SERVICE-CATEGORY> یک رشته که نوع وسیله‌ای که برای انجام این سفارش نیاز است قرار دارد و مقدار آن برابر BIKE یا VAN یا TRUCK است. منظور از <START-POSITION> موقعیت مکانی فرستنده سفارش است، همچنین منظور از <FINISH-POSITION> موقعیت مکانی گیرنده سفارش است. هر دو این مقادارها به صورت یک زوج مرتب از اعداد صحیح به همان روش توضیح داده شده آمده است.

برای مثال یک نمونه از این درخواست به صورت زیر خواهد بود:

```
CREATE-ORDER TRUCK (-100, 100) (100, -100)
```

بعد از دریافت این درخواست، در صورتی که <START-POSITION> برابر <FINISH-POSITION> باشد عبارت `invalid order` و در غیر این صورت شماره این سفارش را برای این سفارش ثبت و سپس چاپ کنید.

هزینه این سفارش ( `COST` ) در این لحظه، از فرمول توضیح داده شده محاسبه می‌شود. همچنین وضعیت ( `STATUS` ) یک سفارش بلافاصله بعد از ایجاد `PENDING` خواهد بود.

#### ▼ درخواست `ASSIGN-NEXT-ORDER`

این درخواست به معنی تقاضا یک راننده برای تخصیص به نزدیک‌ترین سفارش است.

`ASSIGN-NEXT-ORDER <USERNAME>`



در صورتی که راننده‌ای به چنین اسمی نداریم عبارت `invalid driver name` را چاپ کنید. در صورتی که وضعیت این راننده `BUSY` است عبارت `driver is already busy` را چاپ کنید. در صورتی که سفارشی در وضعیت `PENDING` با وسیله یکسان با راننده وجود ندارد عبارت `there is no order right now` را چاپ کنید.

در صورتی که هیچ کدام از حالت‌های بالا پیش نیامد سفارشی با وضعیت `PENDING` را انتخاب می‌کنیم که کمترین فاصله با موقعیت مکانی فعلی راننده را دارد. اگر چند سفارش با فاصله برابر وجود داشتند سفارشی را انتخاب می‌کنیم که زودتر به سیستم اضافه شده است. سپس عبارت زیر را چاپ می‌کنیم.

`<ORDER-ID> assigned to <USERNAME>`



که `ORDER-ID` به معنی شماره سفارش تخصیص یافته به راننده با نام `USERNAME` است. بعد از این تخصیص وضعیت ( `STATUS` ) راننده به `BUSY` و وضعیت ( `STATUS` ) سفارش به `ARRIVED` تغییر خواهد کرد.

#### ▼ درخواست `GET-DRIVER`

در این درخواست وضعیت یک راننده پرسیده می‌شود.



GET-DRIVER <USERNAME>

در صورتی که راننده‌ای به چنین اسمی نداریم عبارت `invalid driver name` را چاپ کنید. در غیر این صورت، ابتدا `STATUS` این راننده را چاپ کنید (یعنی در صورت مشغول بودن رشته `BUSY` و در غیراین صورت رشته `FREE` را چاپ کنید). سپس با یک فاصله موقعیت مکانی فعلی راننده را چاپ کنید سپس با یک فاصله اعتبار حساب راننده را چاپ کنید.

درخواست ORDER-UPDATE ▼

در این دستور یک راننده تغییر وضعیت سفارش خود را اعلام می‌کند.

ORDER-UPDATE <STATUS> <DRIVER-USERNAME> <ORDER-ID>

در صورتی که راننده‌ای به چنین اسمی در سیستم وجود ندارد عبارت `invalid driver name` را چاپ کنید. در صورتی که شماره سفارش تخصیص یافته به این راننده برابر `ORDER-ID` نیست عبارت `wrong order-id` را چاپ کنید. در صورتی که تغییر وضعیت قابل قبول نیست عبارت `invalid status` و در غیر این صورت عبارت `status changed successfully` را چاپ کنید.

توجه کنید وضعیت یک سفارش به صورت زیر تغییر می‌کند:

PENDING -> ARRIVED -> PICKUP -> DELIVERED

پس تغییر وضعیت، قابل قبول است اگر به **دقیقاً به مرحله بعدی** برود. (به طور مثال نباید وضعیت یک سفارش در یک مرحله از `PENDING` به `PICKUP` تغییر کند یا نباید وضعیت یک سفارش به وضعیت فعلی خودش تغییر کند.)

زمانی که وضعیت یک سفارش به حالت `DELIVERED` تغییر می‌کند. 80 درصد هزینه سفارش به حساب راننده اضافه می‌شود و 20 درصد باقی مانده به حساب شرکت اضافه می‌شود. (تضمین می‌شود هزینه‌ای که باید به حساب راننده یا شرکت اضافه شود عدد صحیح است.)

در صورتی که وضعیت سفارش به **PICKUP** تغییر کند موقعیت مکانی راننده به موقعیت مکانی فرستنده سفارش تغییر می‌کند.

در صورت تغییر وضعیت سفارش به **DELIVERED** وضعیت راننده به **FREE** و موقعیت مکانی راننده به گیرنده سفارش تغییر می‌کند.

#### ▼ درخواست GET-ORDER

در این درخواست وضعیت یک سفارش را پیگیری می‌کنیم.

GET-ORDER <ORDER-ID>



در صورتی که چنین سفارشی وجود ندارد عبارت **invalid order** و در غیر این صورت در یک خط ابتدا وضعیت ( **STATUS** ) این سفارش سپس با یک فاصله نام ( **USERNAME** ) راننده‌ای که در حال ارسال این سفارش است و در نهایت با یک فاصله، هزینه ( **COST** ) ارسال این سفارش را چاپ کنید.

اگر این سفارش به هیچ راننده‌ای تخصیص پیدا نکرده است عبارت **None** را به جای نام راننده چاپ کنید.

#### ▼ درخواست GET-ORDER-LIST

در این درخواست لیست تمام سفارش‌هایی که وضعیتشان برابر **<STATUS>** است را چاپ کنید.

GET-ORDER-LIST <STATUS>



شماره تمام سفارش‌هایی که وضعیت ( **STATUS** ) آن‌ها برابر **<STATUS>** است را در یک خط با یک فاصله بین شماره‌ها چاپ کنید. اگر هیچ سفارشی در این وضعیت نبود کلمه **None** را چاپ کنید.

#### ▼ درخواست GET-DRIVER-LIST

در این درخواست لیست تمام راننده‌هایی که وضعیتشان برابر **<STATUS>** است را چاپ کنید.

GET-DRIVER-LIST <STATUS>





نام تمام راننده‌هایی که وضعیت ( STATUS ) آن‌ها برابر <STATUS> است را با یک فاصله بین نام‌ها در یک خط چاپ کنید. اگر هیچ راننده‌ای در این وضعیت نبود کلمه None را چاپ کنید.

#### ▼ درخواست GET-NEAR-DRIVER

در این درخواست COUNT تا نزدیک ترین راننده به نقطه <POSITION> با وضعیت FREE را می‌پرسیم.

GET-NEAR-DRIVER <POSITION> <COUNT>



این لیست را به ترتیب فاصله چاپ کنید (یعنی نام راننده‌ای که فاصله‌ی کمتری با نقطه‌ی موردنظر دارد، باید قبل از نام راننده‌ای که فاصله‌ی بیشتری با نقطه‌ی موردنظر دارد، چاپ شود). اگر دو راننده فاصله یکسانی داشتند راننده‌ای که زودتر به سیستم اضافه شده را زودتر چاپ کنید.

در صورتی که تعداد راننده‌ها کمتر از COUNT بود همه راننده‌های قابل قبول را چاپ کنید. در صورت خالی بودن لیست None چاپ کنید.

#### ▼ درخواست GET-CNT-ORDER

در این درخواست تعداد سفارشات که محل فرستنده یا گیرنده (بستگی به START یا FINISH بودن درخواست) که از نقطه POSITION فاصله حداکثر DISTANCE دارند را چاپ کنید.

GET-CNT-ORDER <POSITION> <DISTANCE> <START/FINISH>



دقت کنید که <START/FINISH> ، همواره یکی از مقادیر START یا FINISH را داراست.

#### ▼ درخواست GET-NEAREST-PENDING-ORDER

شماره ( ORDER-ID ) نزدیک ترین سفارش با وضعیت PENDING که فرستنده آن به POSITION است را چاپ کنید.

GET-NEAREST-PENDING-ORDER <POSITION>



اگر چنین سفارش وجود نداشته کلمه None را چاپ کنید.

در چنین دسترسی وجود ندارد. و در حیرت این صورت دستور دسترسی را چاپ کنید.

#### درخواست GET-COMPANY ▼

در این درخواست سود شرکت را از سفارش‌های تحویل داده شده تا این لحظه چاپ کنید.

GET-COMPANY



پاسخ همواره به صورت یک عدد صحیح است.

## ورودی

در هر سطر از ورودی یکی از انواع درخواست‌های تعریف شده داده می‌شود. قدرمطلق همه اعداد داده شده حداکثر 1000 و تعداد درخواست‌ها در این سوال حداکثر 100 خواهد بود.

## خروجی

در هر سطر خروجی عبارت متناسب با درخواست‌ها را به ترتیب چاپ کنید. در آخرین خط ورودی تنها کلمه END برای نشان دادن پایان دستورات نوشته می‌شود.

## مثال

- توجه کنید که خروجی خط `i`ام، مربوط به ورودی خط `i`ام می‌باشد.

## ورودی نمونه ۱

```
1 ADD-DRIVER Ali (10, 10) VAN
2 ADD-DRIVER GHOLI (0, 0) BIKE
3 ADD-DRIVER Ali (45, 90) VAN
4 CREATE-ORDER VAN (18, 10) (-33, 66)
5 ASSIGN-NEXT-ORDER Ali
6 GET-DRIVER GHOLI
7 GET-DRIVER Ali
8 ORDER-UPDATE DELIVERED ali 1
```



```

9 ORDER-UPDATE PICKUP Ali 1
10 ORDER-UPDATE DELIVERED Ali 1
11 GET-ORDER 1
12 GET-ORDER-LIST ARRIVED
13 GET-ORDER-LIST DELIVERED
14 GET-DRIVER-LIST FREE
15 GET-NEAR-DRIVER (5, -21) 3
16 GET-CNT-ORDER (-40, -80) 100 START
17 GET-CNT-ORDER (0, 0) 100 FINISH
18 GET-NEAREST-PENDING-ORDER (20, 20)
19 GET-COMPANY
20 END

```

خروجی نمونه ۱

```

1 user added successfully
2 user added successfully
3 user previously added
4 1
5 1 assigned to Ali
6 FREE (0, 0) 0
7 BUSY (10, 10) 0
8 invalid driver name
9 status changed successfully
10 status changed successfully
11 DELIVERED Ali 10800
12 None
13 1
14 Ali GHoLI
15 GHoLI Ali
16 0
17 1
18 None
19 2160

```

ورودی نمونه ۲

```
ADD DRIVER M. Hossain (1001, (140, -21), TRUCK
```

```

1 ADD-DRIVER Monammad021 (-142, 21) TRUCK
2 ADD-DRIVER kami (20, 20) VAN
3 ADD-DRIVER 00MOHSEN (0, 128) TRUCK
4 ADD-DRIVER kami (30, 30) BIKE
5 ADD-DRIVER SaMi (55, -2) BIKE
6 GET-DRIVER 00MOHSEN
7 CREATE-ORDER TRUCK (0, 0) (100, 18)
8 CREATE-ORDER BIKE (19, -38) (-40, 127)
9 CREATE-ORDER VAN (16, 16) (-75, 75)
10 ASSIGN-NEXT-ORDER SaMi
11 CREATE-ORDER TRUCK (-10, 100) (-11, 100)
12 ASSIGN-NEXT-ORDER 00MOHSEN
13 ASSIGN-NEXT-ORDER SaMi
14 GET-DRIVER sami
15 GET-DRIVER SaMi
16 ORDER-UPDATE DELIVERED 00MOHSEN 3
17 ORDER-UPDATE PICKUP 00MOHSEN 4
18 ORDER-UPDATE DELIVERED 00MOHSEN 4
19 GET-DRIVER 00MOHSEN
20 GET-ORDER 2
21 GET-ORDER 3
22 GET-ORDER-LIST PENDING
23 GET-ORDER-LIST ARRIVED
24 GET-DRIVER-LIST BUSY
25 GET-DRIVER-LIST FREE
26 GET-NEAR-DRIVER (78, 29) 2
27 GET-NEAR-DRIVER (-70, 50) 1
28 GET-CNT-ORDER (0, 0) 133 FINISH
29 GET-CNT-ORDER (-29, 101) 77 START
30 GET-NEAREST-PENDING-ORDER (17, 23)
31 GET-NEAREST-PENDING-ORDER (0, 0)
32 GET-COMPANY
33 ORDER-UPDATE PICKUP SaMi 2
34 ORDER-UPDATE DELIVERED SaMi 2
35 GET-COMPANY
36 END

```

خروجی نمونه ۲

1 | user added successfully



```
1 user added successfully
2 user added successfully
3 user added successfully
4 user previously added
5 user added successfully
6 FREE (0, 128) 0
7 1
8 2
9 3
10 2 assigned to SaMi
11 4
12 4 assigned to 00MOHSEN
13 driver is already busy
14 invalid driver name
15 BUSY (55, -2) 0
16 wrong order-id
17 status changed successfully
18 status changed successfully
19 FREE (-11, 100) 240
20 ARRIVED SaMi 22500
21 PENDING None 15100
22 1 3
23 2
24 SaMi
25 Mohammad021 kami 00MOHSEN
26 kami 00MOHSEN
27 Mohammad021
28 2
29 1
30 3
31 1
32 60
33 status changed successfully
34 status changed successfully
35 4560
```