

## سوال سوم

در یک سینما N ردیف صندلی داریم. در هر ردیف صندلی‌ها با شماره‌های ۱ تا ۱۰ (از چپ به راست) شماره‌گذاری شده‌اند.

هر ردیف سه جایگاه جدا دارد. یک جایگاه در سمت چپ که ۳ صندلی، جایگاه میانی که ۴ صندلی و جایگاه سمت راست که ۳ صندلی دارد.

در ادامه Q گروه دوست به ترتیب وارد سینما می‌شوند، هر گروه تعداد دلخواهی عضو دارد (فرض کنید iامین گروهی که وارد می‌شود t عضو دارد)

هر گروه در نظر دارد در جای مشخصی از سینما بنشینند و از آن جایی که اعضای هر گروه با هم دوست هستند، تمایل دارند در یک ردیف کنار هم بنشینند. حال شما با توجه به وضعیت نشستن گروه‌های قبلی باید مشخص کنید آیا یک گروه که به تازگی وارد سینما شده است، میتواند در جای دلخواهش بنشیند یا خیر.

مثلا با ورود یک گروه ۳ نفره و درخواست نشستن در ردیف ۳ جایگاه ۴ تا ۶ شما باید با توجه به وضعیت نشستن گروه‌هایی که قبلا وارد شده‌اند مشخص کنید آیا این درخواست معتبر است یا خیر.

کلاس‌هایی که باید پیاده سازی کنید:

- کلاس صندلی که شامل:

- یک متغیر boolean private است که در صورتی که صندلی اشغال باشد، true و در غیر این صورت false است.
- همچنین باید شامل Getter و Setter برای دسترسی و تغییر متغیرها باشد.

- کلاس ردیف که شامل:

- آرایه از صندلی‌ها است (با سطح دسترسی private). هر آرایه متناظر با یک جایگاه (چپ، میانی یا راست) است و سائز آرایه متناسب با ظرفیت هر جایگاه که در ابتدا سوال مشخص شده است تعیین می‌شود.
- Constructor برای ساختن شیء ردیف.
- متد getChair که یک متغیر int مانند c را ورودی می‌گیرد و وضعیت صندلی cام در ردیف را به صورت boolean باز می‌گرداند.
- متد setChair که یک متغیر int مانند c را ورودی می‌گیرد و وضعیت صندلی cام در ردیف را تغییر می‌دهد (از true به false و از false به true).

• کلاس سینما که شامل:

- متغیر rows از نوع int که نشان دهنده تعداد ردیف های سالن سینما است.
- آرایه با طول برابر با rows از ردیف ها است (با سطح دسترسی private).
- Constructor که یک متغیر int مانند rows را می گیرد و آرایه را می سازد.
- متد getChair که دو متغیر int مانند r و c را ورودی می گیرد و وضعیت صندلی cam در ردیف r را به صورت boolean باز می گرداند.
- متد setChair که دو متغیر int مانند r و c را ورودی می گیرد و وضعیت صندلی cam در ردیف r را تغییر می دهد (از true به false و از false به true).
- متد checkChairs که سه متغیر int مانند l, r و x را می گیرد که x شماره ردیف مورد نظر، l چپ ترین صندلی درخواست شده توسط گروه و r شماره راست ترین صندلی درخواست شده توسط گروه است. خروجی متد یک boolean که در صورتی که این گروه بتوانند در آن باز از صندلی ها بنشینند true و در غیر این صورت false است.
- متد occupyChairs که سه متغیر int مانند l, r و x را گرفته، وضعیت صندلی های متناظر را به اشغال شده تغییر می دهد. x شماره ردیف مورد نظر، l چپ ترین صندلی درخواست شده توسط گروه و r شماره راست ترین صندلی درخواست شده توسط گروه است.

• کلاس Main شامل:

- منطق کلی اجرای برنامه و ورودی گرفتن و خروجی دادن.

در خط اول ورودی متغیرهای Q و N وارد می شوند. در Q خط بعدی، درخواست گروه ها بیان می شود. فرمت ورودی:

```
Q N
l1 r1 x1
l2 r2 x2
...
lQ rQ xQ
```

خروجی شامل Q خط، که هر کدام نشان دهنده درخواست گروه متناظر برای دریافت صندلی است.

مثال اول:

```
4 1
1 3 1
output: true
2 6 1
output: false
3 7 1
output: false
4 8 1
output: true
```

مثال دوم:

```
3 2
1 10 0
output: false
4 4 1
output: true
7 8 10
output: false
```

مثال سوم:

```
5 3
1 10 2
output: true
4 7 1
output: true
1 4 3
output: true
7 10 3
output: true
3 9 3
output: false
```

## سوال چهارم

### Number guessing game

در این بازی، کاربر باید عددی که کامپیوتر تعیین کرده است را حدس بزند. بدین صورت که کامپیوتر یک عدد تصادفی در محدوده ۱ تا ۱۰۰ تولید می‌کند، سپس کاربر عدد را حدس می‌زند.

کامپیوتر می‌گوید که آیا عدد تولید شده با عددی که کاربر حدس زده مطابقت دارد یا خیر، و اگر مطابقت نداشت، بیشتر یا کمتر بودن آن را مشخص می‌کند. حدس زدن کاربر تا ۷ بار ادامه می‌یابد، اگر کاربر نتوانست عدد درست را حدس بزند، بازی پایان می‌یابد.

### شروع بازی

هنگام شروع بازی در ابتدا عبارت زیر چاپ می‌شود:

**Enter a number between 1 and 100:**

سپس کاربر عددی که حدس زده را وارد میکند، در صورت درست بودن عدد، عبارت

**You guessed the number!**

در صورت بیشتر بودن عدد نسبت به عدد تولیدی کامپیوتر، عبارت

**Your guess is too high.**

و در صورت کمتر بودن عدد نسبت به عدد تولیدی کامپیوتر، عبارت

**Your guess is too low.**

چاپ می‌شود.

اگر کاربر موفق نشد پس از ۷ بار عدد درست را حدس بزند، عبارت زیر چاپ می‌شود:

**You have used up all your guesses.**

توجه کنید که برای حل این سوال باید از کلاس‌های Main و GuessingGame استفاده کنید و در صورت پیاده سازی کد در یک کلاس، نمره کامل تعلق نمی‌گیرد.

نمونه بازی:

Enter a number between 1 and 100:

50

Your guess is too low.

75

Your guess is too low.

79

Your guess is too high.

77

You guessed the number!

## سوال پنجم (پیاده‌سازی یک کافه)

منوی ورود:

ثبت نام

ورود مشتری

ورود ادمین

خروج

منوی مشتری:

ثبت سفارش

حذف سفارش

بازگشت

منوی ادمین:

نمایش لیست مشتری‌ها و سفارشات آن‌ها

بازگشت

ابتدای برنامه منوی ورود نشان داده می‌شود و کاربر می‌تواند یکی از سه گزینه را انتخاب کند: با انتخاب ثبت نام، اسم کاربر و نوع آن (client و یا admin) به عنوان ورودی دریافت می‌شود و با توجه به IDهای قبلی یک ID یکتا به او اختصاص یافته، و نمایش داده می‌شود. با انتخاب ورود مشتری، از کاربر ID گرفته می‌شود و در صورت وجود داشتن مشتری، منوی مشتری نشان داده می‌شود، در غیر این صورت به منوی ورود برمی‌گردد. (کاربر ادمین نیز امکان ورود به این بخش و ثبت سفارش را دارد) با انتخاب ورود ادمین، ID گرفته می‌شود و در صورت ادمین بودن کاربر، منوی ادمین نشان داده می‌شود.

منوی مشتری:

در بخش ثبت سفارش هر خوراکی با قیمت آن مشخص می‌شود و بعد از انتخاب وارد لیست سفارش‌های مشتری می‌شود و دوباره منو نمایش داده می‌شود. در بخش حذف لیستی از سفارشات فرد نشان داده می‌شود و مورد انتخاب شده در آخر حذف می‌شود. منوی ادمین: در بخش لیست تمام مشتری‌ها می‌توانید لیست تمام مشتری‌ها (مشتری‌های دارای سفارش فعال) و سفارشات آن‌ها را مشاهده کنید. و با انتخاب گزینه‌ی back به منوی ادمین برگردید.

راهنمایی:

در بخش inpuhandler یک متد run تعریف شده است. با استفاده از آن، ابتدا منوی ورود را نشان دهید و ورودی مناسب را دریافت کنید. سپس با توجه به ورودی، متد مناسب را صدا بزنید و در همان متدها ورودی مناسب را دریافت کنید. می‌توانید برای پیاده‌سازی لیست‌های مورد نظر، از آرایه‌ای به طول مشخص استفاده کنید. در صورت نیاز می‌توانید فیلدها و متدهای دیگری را نیز به کلاس‌ها اضافه کنید.

با توجه به ترتیب ورود کاربران، به ترتیب به هریک از آنان یک ID داده شود. به اولین کاربر ID برابر با ۱، به بعدی ۲ و به همین ترتیب تا آخر.

امتیازی:

برای لیست‌های موجود باید یک لینکدلیست پیاده سازی کنید و از آن استفاده کنید. لینکدلیست باید قابلیت حذف و اضافه داشته باشد.

کلاس‌ها:

Client:

```
String name;  
String id;  
DrinkLinkedList orders;
```

Methods:

```
void orderDrink(Drink newOrder);  
void cancelOrder(Drink targetOrder);  
DrinkLinkedList getOrders();
```

Drink:

```
String name;  
double price;
```

Café:

```
Client[] admins;  
Client[] clients;  
Drink[] drinks;
```

Methods:

```
void addClient(Client newClient);  
void removeClient(Client targetClient);  
void showCafeMenu();  
Client getClient(String Client id);  
ClientLinkedList getClients();  
Void updateDrinkOrderNumber();  
int getOrderNumber(Drink);
```

DrinkLinkedList

```
Drink value;
```

```
...
```

MenuHandler:

```
String[] firstMenu;  
String[] clientMenu;  
String[] adminMenu;
```

Methods:

```
Void printFirstMenu();  
Void printClientMenu();  
Void printAdminMenu();  
Void printClients(Café targetCafe);  
Void printAllOrders();  
Void printClientAllOrders(String clientId);
```

InputHandler:

MenuHandler menu;

Cafe cafe;

Client client;

Void registerClient(String name, String id);

Void signInClient(String id);

Void signInAdmin(String id);

Void cancelOrder(int index);

Void addOrder(int index);

Void showToMenu(String[] targetMenu);

Void showClientsList();

Void run();

نمونه اجرا:

output:

```
Signup
Login Client
Login Admin
Exit
```

input:

```
1
```

output:

```
Enter your name and role:
```

input:

```
ShalqamZadeh, client
```

output:

```
Welcome ShalqamZadeh Jaan!
Your id is: 1
```

output:

```
Signup
Login Client
Login Admin
Exit
```

input:



2

output:

Enter your ID:

input:

1

output:

Order drink

Cancel order

Log out

input:

1

با انتخاب این گزینه، نوشیدنی‌های موجود نشان داده می‌شوند:

output:

Tea 45,000

Milk 38,000

Coffee 99,990

Shake 81,000

Watermelon juice 16,000

input:

5

output:

Order drink

Cancel order

Log out

input:

1

output:

Tea 45,000

Milk 38,000

Coffee 99,990

Shake 81,000

Watermelon juice 16,000

input:

3

output:

Order drink

Cancel order

Log out

input:

2

output:

Watermelon juice

Coffee

Back

input:

2

output:

Order drink

Cancel order

Log out

input:

3

output:

Signup  
Login Client  
Login Admin  
Exit

input:

2

output:

Enter your ID:

input:

1

output:

Order drink  
Cancel order  
Log out

input:

3

output:

Signup  
Login Client  
Login Admin  
Exit

input:

1

output:

Enter your name and role:

input:

```
cafeadmin, admin
```

output:

```
Welcome cafeadmin Jaan!  
Your id is: 2
```

output:

```
Signup  
Login Client  
Login Admin  
Exit
```

input:

```
3
```

output:

```
Enter your ID:
```

input:

```
2
```

output:

```
ShalqamZadeh - 1:  
    [Watermelon juice]
```

```
Back
```