

## برنامه‌سازی پیشرفته

### تمرین کامپیوتری شماره ۲



مدرس: رامتین خسروی

طراحان: امیرحسین عارف‌زاده، مهرداد لیویان، پارسا  
سعیدنیا، پریا پاسه‌ورز، فرشته باقری، طاها مجلسی،  
صادق صمدی، سید حمید محمودی، مهدی نائینی

مهلت تحویل: دوشنبه ۲۰ اسفند ۱۴۰۳، ساعت ۲۳:۵۹

## مقدمه

هدف از این تمرین آشنایی شما با طراحی برنامه است. با توجه به حجم تمرین توصیه می‌شود این روال را برای انجام آن طی کنید.

1. قبل از شروع پیاده‌سازی زمانی را به خواندن کامل شرح مسئله اختصاص دهید.
  2. پس از آن به نحوه شکست برنامه به اجزای کوچکتر (مثل روال‌های خواندن ورودی اولیه و رسیدگی به دستورات کاربر) فکر کنید.
  3. بر اساس شکست بند قبل، درباره تایپ‌هایی که اطلاعات ثابت برنامه (در اینجا اطلاعات میزها و دانشجویان) و همچنین وضعیت برنامه (وضعیت جاری میزها) را نگه می‌دارند تصمیم‌گیری کنید.
  4. توصیه می‌شود به این ترتیب پیاده‌سازی خود را پیش ببرید:
    - a. ابتدا خواندن فایل‌های ورودی را کامل کنید.
    - b. دستورها را یکی یکی در برنامه خود پیاده‌سازی کنید.
    - c. بعد از پیاده‌سازی هر دستور درستی کارکرد برنامه خود را با داده‌های آزمون بیازمایید.
- به این ترتیب، حتی اگر فرصت نکردید تمام برنامه را پیاده‌سازی کنید آنچه تا پایان مهلت تمرین دارید برنامه‌ای است که کار می‌کند (هر چند کامل نیست).
- برای آشنایی بیشتر با مفاهیم طراحی بالا به پایین می‌توانید به ویدیویی که در بخش محتوای دستیاران آموزشی در [صفحه درس](#) یا [وبسایت](#) درس مراجعه کنید.

# مدیریت فضای کار اشتراکی

در این برنامه، شما وظیفه دارید یک سیستم مدیریت فضای کار اشتراکی (مثل سایت دانشکده) را پیاده‌سازی کنید. این سیستم با ورود هر دانشجو، میزهای مناسب را به او پیشنهاد می‌دهد و او می‌تواند از میان آن‌ها یک میز را انتخاب کند. همچنین، هنگام خروج او، افرادی که در صف انتظار آن میز هستند را بررسی می‌کند. سیستم با توجه به اولویت‌های دانشجویان و روابط دوستی (یا دشمنی) بین آنها سعی می‌کند بهترین پیشنهاد را ارائه کند.

## قالب فایل‌های ورودی

اطلاعات مربوط به میزها و دانشجویها در دو فایل جداگانه قرار دارند که باید اطلاعات موجود در آن‌ها را در ابتدای اجرا برنامه خوانده و ذخیره کنید. نوع فایل‌های ورودی به صورت CSV<sup>1</sup> هستند و برای آشنایی با این نوع فایل‌ها می‌توانید [این لینک](#) را بررسی کنید.

**تذکر:** تضمین می‌شود داده‌های تمام فایل‌های CSV مطابق توضیحات می‌باشند.

در ادامه شرح هر یک از فایل‌ها آمده است. هر فایل csv یک سطر header در ابتدای خود دارد که نام هر یک از ستون‌ها را به ترتیب مشخص می‌کند.

## فایل میزها

در این فایل پنج ستون شناسه، مختصات (x, y)، ظرفیت و نوع میز وجود دارد. نوع میزها به سه صورت مشخص می‌شود:

- 1: میزهای نزدیک پنجره (window)
- 2: میزهای نزدیک در خروج (door)
- 3: میزهای وسط سایت (middle)

نام ویژگی	توضیحات	نوع داده
ID	شناسه میز	عدد طبیعی
X	مختصات طولی	عدد طبیعی
Y	مختصات عرضی	عدد طبیعی

<sup>1</sup> Comma-Separated Values

Capacity	ظرفیت میز	عدد طبیعی
Type	نوع میز	رشته

نمونه فایل میزها
Id,x,y,capacity,type 1,2,3,4>window 2,4,5,8,door 7,12,6,1,middle

### فایل دانشجوها

در این فایل چهار ستون شناسه، اسم، شناسه دوست و شناسه دشمن وجود دارد.

توجه: دوستی‌ها و دشمنی‌ها روابط دو طرفه نیستند. (به عنوان مثال اگر علی دوست رضا باشد، لزوماً به این معنا نخواهد بود که رضا هم دوست علی است)

توجه: تضمین می‌شود که هر کس دقیقاً یک دوست و یک دشمن دارد، کسی دوست یا دشمن خودش نیست و دوست و دشمن هر کس دو فرد متفاوت هستند.

نوع داده	توضیحات	نام ویژگی
عدد طبیعی	شناسه دانشجو	ID
رشته	نام دانشجو	Name
عدد طبیعی	شناسه دوست	Friend_ID
عدد طبیعی	شناسه دشمن	Enemy_ID

نمونه فایل دانشجو ها
id,name,friend_id,enemy_id 1,Paria,2,4

4, Parsa, 2, 1  
2, Mehrad, 4, 1

## نحوه اجرای برنامه

در زمان اجرای برنامه مسیر فایل‌های ورودی به ترتیب (ابتدا مسیر فایل میزها، سپس مسیر فایل دانشجویها) از طریق آرگومان خط فرمان به برنامه داده می‌شود. برای آشنایی با این نوع آرگومان‌ها، می‌توانید [این لینک](#) را مشاهده کنید.

### قالب ورودی

```
./SiteManager.out </path/to/tables/file> </path/to/students/file>
```

## دستورات برنامه

### دستور نمایش اطلاعات میز

این دستور با گرفتن شناسه میز، اطلاعات مربوط به میز را نمایش می‌دهد.

این اطلاعات شامل شناسه، اسامی افراد حاضر **مرتب شده** به ترتیب صعودی، ظرفیت باقی‌مانده و تعداد افراد حاضر در صف انتظار میز می‌باشد.

### قالب دستور

```
show_table_info <Table_ID>
```

### قالب خروجی

```
Table ID: <Table_ID>  
People at the table: <People_At_The_Table>  
Table remaining capacity: <Remaining_Capacity>  
Waiting queue length: <Waiting_Queue Length>
```

تضمین می‌شود شناسه‌های وارد شده جزو شناسه‌های موجود می‌باشد و خطایی در این قسمت رخ نمی‌دهد.

نمونه ورودی
show_table_info 1
نمونه خروجی
Table ID: 1 People at the table: Mehrad, Paria, Parsa Table remaining capacity: 1 Waiting queue length: 0

## دستور ورود دانشجو

در این دستور، بعد از وارد کردن شناسه دانشجو توسط کاربر، اطلاعات تمام میزها بر اساس اولویتشان چاپ می‌شوند.

تضمین می‌شود شناسه دانشجو وارد شده توسط کاربر، در فایل CSV دانشجوها وجود دارد و دانشجو پیش از این وارد سایت نشده است.

قالب دستور
enter <Student_ID>
قالب خروجی
Table <Table_ID_1>: <Remaining_Capacity_1> <Waiting_Queue_Length_1> Table <Table_ID_2>: <Remaining_Capacity_2> <Waiting_Queue_Length_2> ...

پس از وارد کردن این دستور، اطلاعات تمام میزها بر اساس اولویتشان به صورت نزولی نمایش داده می‌شوند. این اطلاعات شامل شناسه، ظرفیت باقی‌مانده و تعداد افراد منتظر در صف انتظار میز است.

توجه داشته باشید که اولویت هر میز بر اساس دوستی و دشمنی‌ها مشخص می‌شود که جلوتر **نحوه محاسبه اولویت** گفته خواهد شد.

نمونه ورودی
enter 1
نمونه خروجی
Table 1: 4 0 Table 2: 8 0 Table 7: 0 1

## دستور رزرو میز

پس از مشاهده اطلاعات تمام میزها، کاربر با وارد کردن این دستور، میز مورد نظر را رزرو می‌کند.

**توجه کنید که ممکن است این دستور بدون وارد کردن دستور ورود دانشجو وارد شود و الزامی نیست همواره پشت سر هم وارد شوند.**

این دستور شامل یک آرگومان اجباری که شناسه دانشجو است و یک آرگومان اختیاری که شناسه میز مورد نظر است، می‌باشد.

در صورت وارد نشدن شناسه میز باید بهترین میز را برای دانشجو انتخاب کنید. بهترین میز صرف نظر از پر بودن یا ظرفیت داشتن انتخاب می‌شود.

اگر میز مورد نظر ظرفیت باقی‌مانده داشت، سر آن میز می‌نشیند، در غیر این صورت فرد در صف انتظار آن میز قرار می‌گیرد. (ظرفیت صف انتظار میزها نامحدود است).

تضمین می‌شود شناسه دانشجو وارد شده توسط کاربر، متعلق به دانشجویی که قبلاً سر میزی نشسته است، نیست.

قالب دستور
reserve_table <Student_ID> <Table_ID>
قالب خروجی
<Student_Name> enters the waiting queue of table <Table_ID>   <Student_Name> sits at table <Table_ID>

پس از وارد کردن دستور، بر اساس پر بودن یا ظرفیت خالی داشتن میز، پیغامی متناسب با آن نمایش داده می‌شود.

تضمین می‌شود شناسه‌های وارد شده جزو شناسه‌های موجود می‌باشد و خطایی در این قسمت رخ نمی‌دهد.

نمونه ورودی ۱
reserve_table 1 2
نمونه خروجی ۱
Paria enters the waiting queue of table 2

نمونه ورودی ۲
reserve_table 1
نمونه خروجی ۲
Paria sits at table 1

## دستور خروج دانشجو

با استفاده از این دستور و وارد کردن شناسه دانشجو، او از سایت خارج می‌شود.

قالب دستور
exit <Student_ID>
قالب خروجی
<Student_Name> exits!

پس از وارد کردن این دستور، پیام خارج شدن دانشجو نمایش داده می‌شود و همچنین در صورت حضور، دانشجوی ابتدای صف انتظار آن میز جایگزین او می‌گردد.

توجه: اگر دوست دانشجو در صف انتظار آن میز حضور داشت، اولویت با جایگزینی او است.

نمونه ورودی
exit 1
نمونه خروجی

Paria exits!

## دستور جابه‌جایی دانشجو

با استفاده از این دستور و وارد کردن شناسه دانشجو، او با دوستش جابه‌جا می‌شود.

توجه : تضمین می‌شود که هم خود دانشجو و هم دوستش سر میزی نشسته‌اند.

قالب دستور
switch <Student_ID>
قالب خروجی
<Student_Name> switches seats with <Friend_Name>!

پس از وارد کردن این دستور، پیام جابه‌جا شدن دانشجو با دوستش نمایش داده می‌شود.

نمونه ورودی
switch 1
نمونه خروجی
Paria switches seats with Mehrad!

## اولویت‌بندی میزها

برای هر میز، امتیازی محاسبه می‌شود که میز با امتیاز بیشتر دارای اولویت بالاتر است.

نحوه محاسبه امتیاز به شرح زیر است:

$$\text{Score} = \text{Distance\_To\_Enemy} - \text{Distance\_To\_Friend} + \text{Type\_Bonus}$$

اگر دشمن یا دوست سر میزی نبودند، فاصله برابر صفر در نظر گرفته می‌شود.

اگر کسی در صف انتظار قرار داشت، در محاسبات تعیین اولویت در نظر گرفته نمی‌شود.

منظور از فاصله در اینجا، فاصله منهتنی (Manhattan Distance) است که به صورت زیر محاسبه می‌شود:

$$\text{Distance} = |x_2 - x_1| + |y_2 - y_1|$$

توجه: در صورت برابر بودن امتیاز دو میز، میز با شناسه کمتر اولویت بیشتری دارد.



امتیازات Bonus برای میزها به صورت زیر است:

Type	Bonus
Window	6
Door	4
Middle	2

## مثال

نمونه ورودی
<pre>enter 1 reserve_table 4 7 enter 2 reserve_table 2 show_table_info 7 exit 4 show_table_info 7</pre>
نمونه خروجی
<pre>Table 1: 4 0 Table 2: 8 0 Table 7: 1 0 Parsa sits at table 7 Table 7: 0 0 Table 2: 8 0 Table 1: 4 0 Mehrad enters the waiting queue of table 7 Table ID: 7 People at the table: Parsa Table remaining capacity: 0 Waiting queue length: 1</pre>

```
Parsa exits!  
Table ID: 7  
People at the table: Mehrad  
Table remaining capacity: 0  
Waiting queue length: 0
```

## نحوه مرتب‌سازی بر اساس نام

در زبان C++، هر رشته شامل تعدادی کاراکتر است و هر کاراکتر دارای یک کد ASCII است. این کد برای حروف انگلیسی کوچک از ۹۷ تا ۱۲۲ و برای حروف انگلیسی بزرگ از ۶۵ تا ۹۰ است. جدول کامل ASCII را می‌توانید در [این لینک](#) مشاهده کنید.

با استفاده از این روش کدگذاری، می‌توانیم یک روش مقایسه رشته‌ها را به صورت مقایسه این کدها با یکدیگر در نظر بگیریم. این روش به این صورت است که کد ASCII کاراکترهای اول دو رشته با همدیگر مقایسه می‌شوند، در صورتی که یکی از آن‌ها کوچک‌تر از دیگری باشد، کل آن رشته کوچک‌تر از رشته دیگر در نظر گرفته می‌شود. در صورتی که کاراکترهای اول دو رشته با همدیگر برابر باشند، کاراکترهای دوم را مقایسه می‌کنیم. این کار را تا زمانی انجام می‌دهیم که به کاراکتری برسیم که در دو رشته یکسان نباشد. در صورتی که پیش از رسیدن به چنین کاراکتری به انتهای هر دو رشته برسیم، دو رشته با همدیگر برابر در نظر گرفته می‌شوند. در زبان C++، [اپراتورهای مقایسه‌ای](#) نیز برای رشته‌ها دقیقاً به همین صورت پیاده‌سازی شده‌اند.

در این تمرین، در بخش‌هایی که خواسته شده آیتم‌ها را بر اساس نام مرتب کنید، لازم است دقیقاً به همین صورت عمل کنید. برای مثال نام‌های زیر را در نظر بگیرید:

نمونه نام‌ها
Amir, amin, Borna, dorsa, AMirreza

در صورت مرتب‌سازی، نام‌ها به صورت زیر خواهند بود:

نمونه نام‌های مرتب‌شده
AMirreza, Amir, Borna, amin, dorsa

## نکات و نحوه تحویل

- تحویل این تمرین در سامانه گیت‌هاب انجام می‌شود. برای انجام تمرین لطفاً از طریق [این لینک](#) وارد شوید، پس از آن باید شماره دانشجویی خود را انتخاب کنید (دقت کنید که با کمک این شماره دانشجویی به شما نمره خواهیم داد، لطفاً در انتخاب درست شماره دانشجویی حتماً دقت کنید، در صورتی که به مشکل خوردید با دستیاران در ارتباط باشید). پس از آن به صفحه‌ای منتقل می‌شوید که در آنجا می‌توانید تمرین جدید را قبول کنید، پس از قبول کردن تمرین یک مخزن<sup>2</sup> در [AP Assignments](#) برای شما ساخته می‌شود و باید کدهای خود را در آنجا قرار دهید. لطفاً دقت کنید که نهایی خود را حتماً در فایل `src/main.cpp` قرار دهید.

- پس از انجام تمرین و بارگذاری در گیت‌هاب، کد Hash آخرین کامیت<sup>3</sup> را به همراه شماره دانشجویی خود در سامانه ایلرن آپلود کنید (در خط اول شماره دانشجویی، پس از آن از `Enter` استفاده کنید و به خط بعد بروید و پس از آن Hash آخرین کامیت). نمونه متن خواسته شده در سامانه ایلرن (بخش `<last_commit_hash>` و `<sid>` را جایگزین کنید):

```
<sid>
<last_commit_hash>
```

نمونه:

```
810100000
bad8fbcdfcf3b9feb371a31e0c370150aa870b18
```

- دقت کنید که عدم رعایت ساختار گفته شده در آپلود یا تغییر ساختار فایل‌ها در مخزن (می‌توانید به دلخواه خود فایل اضافه کنید و ... اما اسم و ساختار فایل‌هایی که در ابتدا به شما داده می‌شود نباید تغییر کند) باعث کسر 5 درصد از نمره شما خواهد شد.
- سعی کنید مرحله به مرحله کد را کامیت کنید و از کامیت کردن کل کد در انتهای پروژه خودداری کنید. همچنین، پیام کامیت‌های شما باید خوانا باشد و تغییر ایجاد شده را توضیح دهد. دقت کنید که بخشی از نمره شما را استفاده صحیح از گیت تشکیل می‌دهد. برای مطالعه در خصوص نحوه صحیح کامیت کردن، می‌توانید [این لینک](#) را مطالعه کنید.
- برنامه شما باید در سیستم عامل لینوکس و با مترجم `g++` با استاندارد `c++20` ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.

---

<sup>2</sup> Repository

<sup>3</sup> Commit

- بخش مهمی از ارزیابی برنامه شما درستی عملکرد آن است. بنابراین به اندازه کافی ورودی‌های آزمایشی طراحی کنید تا درستی خروجی در حالت‌های مختلف آزموده شود و دقت کنید که اگر برنامه شما با مثال‌های داده شده درست کار کند لزوماً به معنای درستی در تمام سناریوها نیست.
- درستی برنامه شما از طریق آزمون‌های خودکار سنجیده می‌شود. به این ترتیب، لازم است خروجی تولید شده از نظر بزرگی و کوچکی حروف، رعایت فاصله‌ها، عدم وجود خروجی‌های اضافه، ... دقیقاً مانند نمونه‌های داده شده باشد. بنابراین پیشنهاد می‌شود که با استفاده از ابزارهایی مانند diff فرمت خروجی برنامه خود را با خروجی‌هایی که در اختیارتان قرار داده شده است مطابقت دهید.
- هدف این تمرین یادگیری شماسست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق سیاست درس با آن برخورد خواهد شد.

## نمرات

- تمیزی کد
  - رعایت کردن نام‌گذاری صحیح و منسجم بودن<sup>4</sup>
  - عدم وجود کد تکراری
  - رعایت دندانه‌گذاری<sup>5</sup>
  - استفاده صحیح از متغیرهای ثابت<sup>6</sup> به جای Magic Value-ها<sup>7</sup>
  - استفاده مناسب از container-ها و iterator-ها به جای روش‌های قدیمی
  - استفاده مناسب از ساختار داده‌های مناسب
  - توابع کوتاه که فقط یک کار را انجام می‌دهند
- درستی کد
  - آزمون‌های خودکار
- طراحی

---

<sup>4</sup> Consistency

<sup>5</sup> Indentation

<sup>6</sup> Constant

<sup>7</sup> به مقادیر خاصی که در کد استفاده می‌شود و برای عملکرد صحیح کد ضروری است اما دلیل استفاده از آن‌ها مشخص نیست و قابل جایگزین شدن با یک ثابت یا اسم مشخص جهت افزایش خوانایی هستند، magic value گفته می‌شود. برای آشنایی بیشتر با این مفهوم می‌توانید [این لینک](#) را مشاهده کنید.

- شکستن مناسب و مرحله به مرحله مسئله
- ذخیره اطلاعات در ساختار داده‌های مناسب
- ساختاردهی کد در قالب توابع کوتاه که فقط یک کار را انجام می‌دهند
- پرهیز از متغیرهای سراسری

#### ● گیت

- پیام‌های خوانا برای commit-ها
- Commit کردن مرحله به مرحله

دقت کنید که موارد ذکر شده لزوماً کل نمره شما را تشکیل نمی‌دهند و ممکن است با تغییراتی همراه باشند.