به نام خدا

درس برنامه نویسی پیشرفته

تمرین دوم

دانشکده مهندسی کامپیوتر

دانشگاه علم و صنعت ایران

استاد مرضیه ملکی مجد

نیم سال دوم ۱۴۰۱-۱۴۰۲

مهلت ارسال :

۱۴۰۱/۱۲/۲۳

مبحث:

آرایه و متد - مقدمات شی گرایی

مسئول تمارین :

آریا شهسوار

فهرست

آداب نامه تمرینات

- پاسخ تمامی سوالات تنها به زبان C# قابل قبول می باشد
- علیرغم اعتماد کامل تیم تی ای به شما دانشجویان عزیز ، تمامی کد های شما با سایر دانشجویان بصورت خودکار و توسط برنامه مقایسه خواهند شد . همچنین در طول ترم ، از تمامی پاسخ های شما ارائه گرفته خواهد شد و نحوه کار تمامی بخش های هر سوال از شما پرسیده خواهد شد ، لذا از کپی نمودن کد دوستانتان خودداری کنید و تمامی پاسخ ها ، کد خودتان باشد . همچنین از آنجایی که مشورت و هم فکری با سایر دوستان بسیار کار پسندیده و مفیدی است - برخلاف کپی کردن کد (: - در صورت هم فکری با دانشجوی (دانشجویان) ، نام وی را بصورت کامنت شده در ابتدای کد خود بنویسید .
- برای ارسال تمارین در طول ترم ، در مجموع ۱۰ روز می توانید تاخیر داشته باشید و در صورتی که جمع تاخیر دانشجویی بیشتر از ۱۰ روز شود ، تمرین وی قابل قبول نخواهد بود لذا تلاش کنید تمرینات را در زمان مقرر در سامانه آپلود کنید
- برای هر تمرین در زمان ددلاین دانشجو باید درخواست خود را مبنی بر ارسال با تاخیر به تی ای ها اعلام نماید
- در تمامی تمرینات سعی شده است که سوالات ساده تر در ابتدا و سوالات دشوار تر در انتهای فایل قرار گیرند (از ساده به دشوار مرتب شده اند)
- در صورت وجود هرگونه سوال در مورد تمرینات ، سعی کنید تا جایی که امکان دارد سوال خود را در گروه بپرسید چرا که شاید سوال شما ، سوال دوستتان نیز باشد و دوستانتان نیز بتوانند از پاسخ سوال شما بهره ببرند .

نکات تمرین سری دوم

- سوالات را در سامانه کوئرا و در قسمت تمرین سری دوم آپلود نمایید .
- دقت شود سوالات از این پس تست کیس نداشته و تشخیص درستی و نمره پاسخ شما با بررسی تست کیس های متفاوت و توضیح راه حلتان در زمان ارائه داده خواهد شد
- برای سوال ۳ و سوال ۵ میتوانید از آرایه و یا لیست استفاده کنید. برای آشنایی بیشتر با لیست می توانید از لینک های زیر استفاده کنید:

https://www.c-sharpcorner.com/article/c-sharp-list

https://www.geeksforgeeks.org/c-sharp-list-class/

- از آنجایی که هر سوال توسط یک تی ای طرح شده است ، تنها تی ای طراح آن سوال می تواند شما را بصورت دقیق راهنمایی کند به همین منظور طراح هر سوال در زیر نوشته شده است تا در صورت ابهام و پرسش در مورد هر سوال ، در صورتی که نیاز به پرسش سوال بصورت انفرادی در پیوی هست ، به تی ای مربوطه مراجعه بفرمایید
  - سوال ۱ . آقای درپوش
  - سوال ۲ . آقای اوشنی
  - سوال ۳ . آقای اوشنی
  - سوال ۴ . آقای درپوش
  - سوال ۵ . آقای ارشیا

تمرین ۱ . کسر لاتکی

صورت این سوال را در صفحه کوئرا تمرین میتوانید مشاهده کنید.

تمرین ۲ . هندسه مقدماتی

در این سوال شما سیستمی برای به دست آوردن اشکال هندسی درست می کنید. این سیستم شامل 3 کلاس است.

۱ . کلاس Point :

۱.۱. شامل دو متغیر خصوصی x و y است که می توانند اعشاری نیز باشند.

۱.۲. شامل یک کانستراکتور دیفالت که مقادیر x و y را برابر 0 قرار می دهد و یک کانستراکتور با دو ورودی دلخواه x و y است.

۱.۳. شامل یک تابع عمومی DistanceTo است که برای ورودی یک نقطه میگیرد و فاصله اقلیدسی نقطه کنونی را تا آن برمیگرداند.

۱.۴. شامل تابع move است که دو ورودی y, x می گیرد و مختصات نقطه را برابر y, x ورودی قرار می دهد.(این تابع خروجی ندارد) public or what?

۲ . کلاس Circle :

۲.۱. شامل دو متغیر خصوصی radius و center است که به ترتیب از جنس double و point هستند.
(توجه شود که مقدار radius در هیچ مرحله ای نمی تواند منفی باشد.)

۲.۲. شامل دو کانستراکتور با ورودی های (radius, center) و(radius, y, x) می باشد.

۲.۳. شامل دو تابع Area و Perimeter است که به ترتیب مساحت و محیط دایره را خروجی می دهند.

۲.۴. شامل تابع GetReletivePosition است که یک نقطه به عنوان ورودی می گیرد موقعیت آن نقطه نسبت به دایره را خروجی می دهد.(اگر توی دایره باشد 1- اگر روی دایره باشد 0 و اگر خارج دایره باشد عدد 1 برگردانده می شود)

توجه : شما باید برنامه خود را جوری بنویسید که اگر در ساخت دایره از کانستراکتور نوع اول استفاده شد با تغییر مکان نقطه center دایره نیز تغییر مکان دهد.

۳ . کلاس Rectangle :

۳.۱. این کلاس شامل 4 نقطه است که راس های مستطیل را مشخص می کنند.

۳.۲. این کلاس شامل دو کانستراکتور است در کانستراکتور اول 4 نقطه (d, c, b,a) که 4 راس مستطیل هستند به شما داده می شود . در کانستراکتور دوم فقط دو نقطه دو سر یک قطر (c, a) به شما داده می شود و شما باید نقاط b و d را به دست بیاورید.

۳.۲.۱. دوسر یک قطر به شما داده می شود و شما باید 2 راس دیگر مستطیل را بدست بیاورید. lack of data --->

۳.۲.۲. 4 راس مستطیل به شما داده می شود . (توجه شود اضلاع مستطیل ورودی حتما با محور های مختصات موازی است) what does it mean? should we check it??

۳.۳. شامل دو تابع Area و Perimeter است که به ترتیب مساحت و محیط دایره را خروجی می دهند.

۳.۴. شامل تابع move است که دو ورودی y, x می گیرد و مستطیل را به اندازه بردار (y, x) جابجا می کند.

تابع Main :

در این تابع شما باید به ازای هر کانستراکتور هر کلاس یک شی بسازید و تمام تابع های موجود را روی آن صدا بزنید (ورودی های تابع ها به صورت دلخواه انتخاب شود .)

تمرین ۳ . شبیه سازی اداره

در این سوال شما باید موارد زیر را طراحی کنید.

کلاس کارمند :

این کلاس شامل فیلد های نام، نام خانوادگی، میزان درآمد، شماره شناسایی و شماره کارمندی است.

همچنین این کلاس دارای متد های employee comparator، find employee، what_is_my_work_ID میباشد.

نکات :

۱. شماره شناسایی هر فرد باید یکتا باشد. بدین منظور شما باید تمام کسانی را که به عنوان کارمند ثبت نام میکنند را در فیلدی درون کلاس ذخیره کنید. در صورت وجود این شماره شناسایی در میان کارمندان ثبت نام شده در خروجی پیغام should we do this in main? we have this ssn right now را چاپ کنید و از ثبت نام او خودداری کنید.

۲. میزان درآمد هر نفر نباید در خارج از کلاس در دسترس باشد.

۳. برای ساختن یک شخص وجود فیلد های نام خانوادگی و میزان درآمد و شماره شناسایی الزامی میباشد.

۴. شماره کارمندی هر شخص در سازنده و با توجه به این که او چندمین نفری است که ثبت نام کرده است ست میشود.

۵. متد find employee در ورودی یک رشته را به عنوان شماره شناسایی دریافت میکند و در صورت وجود کارمندی با این شماره شناسایی او را در خروجی برمی گرداند در غیر این صورت پیام not found را در خروجی چاپ کرده و مقدار null بر میگرداند.

۶. برای استفاده از متد find employee احتیاجی به ساختن شی نیست و این متد بدون داشتن شی و از طریق کلاس کارمند در دسترس است.

۷. با فراخواندن متد what_is_my_work_ID مقدار شماره کارمندی شخص را برمیگردانیم.

۸. در متد comparator دو شماره شناسایی employee در ورودی دریافت می کنیم. در صورت عدم وجود کارمندان با شماره شناسایی های مد نظر در خروجی پیغام Invalid task

نوشته شود. در غیر این صورت اگر میزان درآمد شخص با شماره شناسایی اول بیشتر بود نام و نام خانوادگی او چاپ شود در صورت برابری عبارت equal چاپ شود و در غیر این صورت نام و نام خانوادگی شخص با شماره شناسایی دوم چاپ شود.

۹. برای ذخیره کردن کارمندان ثبت نام شده می توانید از آرایه یا لیست استفاده کنید

تمرین ۴ . مسیر خاص

علی می خواهد در مسابقات رباتیک شرکت کند و مشغول تست ربات خود میباشد. مسیر مسابقه به صورت یک مستطیل شبکه بندی شده است که نقطه ی شروع و پایان و همینطور موانع در آن مشخص است. علی می خواهد بداند که چند راه برای رسیدن از نقطه ی شروع به پایان وجود دارد به صورتی که رباتش دقیقا یک بار از روی تمامی نقاطی که در آن مانعی وجود ندارد، عبور کند.

ورودی :

در خط اول n و m داده میشود که به ترتیب تعداد ردیف ها و ستون ها است.

$$0 \le n,m \le 20$$

در هر n خط بعدی m عدد داریم که میتوانند به صورت زیر باشند:

- 1 : نشان دهنده مربع شروع است. دقیقاً یک مربع شروع وجود دارد.
- 2 : نشان دهنده مربع انتهایی است. دقیقاً یک مربع انتهایی وجود دارد.
- 0 : نشان دهنده مربع های خالی است که می توانیم روی آنها راه برویم.
- -1 : نشان دهنده موانعی است که نمی توانیم از آنها عبور کنیم.

خروجی :

تعداد مسیر های موجود.

Example 1 :

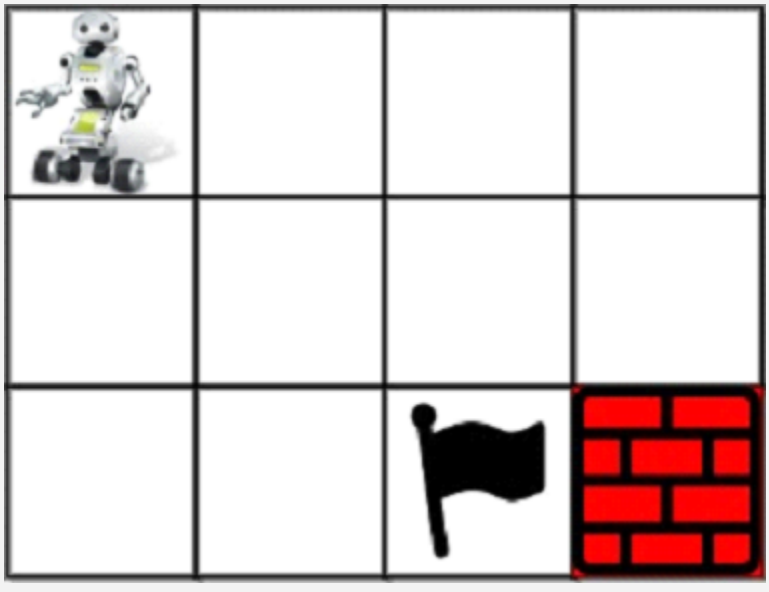
Input :

```
4 3
1 0 0 0
0 0 0 0
0 0 2 -1
```

Output:

```
2
```

دلیل :



ربات دو مسیر زیر را میتواند طی کند :

1.

```
(0,0),(0,1),(0,2),(0,3),(1,3),(1,2),(1,1),(1,0),(2,0),
(2,1),(2,2)
```

2.

```
(0,0),(1,0),(2,0),(2,1),(1,1),(0,1),(0,2),(0,3),(1,3),
(1,2),(2,2)
```

Example 2 :

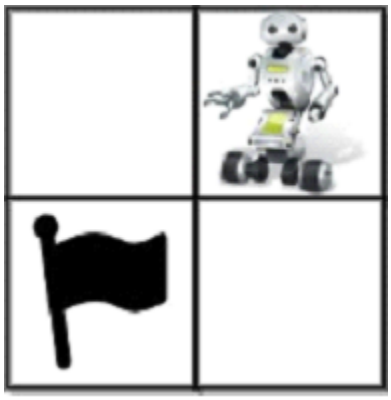
Input :

```
2 2
0 1
2 0
```

Output:

```
0
```

دلیل :



مسیری که تمام نقاط مورد نظر سوال را پوشش دهد وجود ندارد .

توجه : دقت شود که نقاط شروع و پایان می تواند در هر جایی از مستطیل باشند

تمرین ۵ : بورس پیشرفته!

بعنوان یک آنالیزور بازار بورس و سهام قصد داریم تا وضعیت خرید و فروش یک شرکت را تحلیل کنیم .

برای این کار می بایست کلاسی به نام **StockMarket** پیاده سازی کنیم بطوریکه فیلد های زیر را داشته باشد :

۱ . MarketName : نام بازار بورسی

۲ . StockValues[] : ارزش سهم مد نظر در این بازار در یک بازه خاص که بصورت آرایه ای از اعداد صحیح می باشد - در واقع هر المان آرایه بیانگر ارزش سهام در روز i ام می باشد (تضمین می شود که طول این بازه بیشتر از ۱۰۰ نخواهد بود)

و کلاس **Company** شامل فیلدهای زیر:

۱ . CompanyName: نام شرکت

۲ . WorkingMarkets: که می تواند لیست یا آرایه ای از StockMarket باشد که مشخص کننده بازارهایی است که یک شرکت در آن ها فعالیت میکند.

کلاس Company شامل توابع زیر می باشد :

۱ . تابع BuyStockMethods :

این تابع بعنوان ورودی آرایه ای از اعداد صحیح که شامل مقادیری است که شرکت در هر روز دقیقا به همان میزان می تواند خرید ثبت کند را دریافت می کند

در خروجی تمام روش های خرید یک سهم را برمیگرداند - به مثال توجه کنید :

Example 1 :

where do i get this input?

در این مثال موجودی شرکت ۴ میلیون دلار در نظر گرفته شده است

```
TotalCash = 4
```

و به تابع مقادیر زیر بعنوان ورودی داده می شوند :

```
dailyBuys = { 1 , 2 , 3 }
```

change problem

خروجی تابع شامل تمام حالت های تقسیم موجودی با توجه به آرایه ورودی می باشد :

```
{ 1 , 1 , 1 , 1 } , { 1 , 1 , 2 } , { 2 , 2 } , {1,3}
```

نکته ۱ : از آنجایی که برای هر کدام از روش ها ترتیب های متفاوتی وجود دارد ، هر کدام از ترتیب ها را برگردانید تفاوتی ندارد و درست است :

```
{ 1 , 1 , 2 } = { 1 , 2 , 1 } = { 2 , 1 , 1 }
```

نکته ۲ : همچنین ترتیب کلی روش ها نیز مهم نیست :

```
{ 1 , 1 , 2 ) , { 2 , 2 } = { 2 , 2 } , { 1 , 1 , 2 }
```

Example 2 :

در این مثال موجودی شرکت ۱۰ میلیون دلار در نظر گرفته شده است

```
TotalCash = 10
```

و به تابع مقادیر زیر بعنوان ورودی داده می شوند :

```
dailyBuys = { 2 , 5 , 3 , 6 }
```

خروجی تابع شامل تمام حالت های تقسیم موجودی با توجه به آرایه ورودی می باشد :

```
{ 2 , 2 , 2 , 2 , 2 }  , { 2 , 2 , 3 , 3 } ,
{ 2 , 2 , 6 } , { 2 , 3 , 5 } , { 5 , 5 }
```

۲ . تابع BestPeriods :

در این تابع قصد داریم از بین بازارهای بورسی که یک شرکت در آنها فعالیت می کند، بازاری که ممکن است بیشترین سود را برای شرکت داشته باشد پیدا کنیم.

با توجه به وضعیت سهام در بازار هایی که یک شرکت در آنها فعالیت میکند و با توجه به ارزش سهام در یک بازه خاص بیشترین سود ممکن را با انتخاب درست بازه های خرید و فروش دریافت کنیم

در واقع با توجه به فیلد StockValues و تشخیص بهترین زمان برای خرید و فروش سهم ، حداکثر سودی دریافتی را می یابیم.

بدیهی است اگر ارزش سهم در یک بازار همواره کاهش یابد مقدار سود دریافتی صفر خواهد بود

به مثال توجه کنید : like the previous one

Example 1 :

در این مثال آرایه StockValues به صورت زیر است :

```
New York :StockValues[] = {100,180,260,310,40,535,695}

Hong Kong: StockValues[] = {4,2,2,2,4}
```

در بازار اول یکی از راه های ممکن برای اینکه بیشترین سود را دریافت کنیم (ممکن است در مثالی چندین راه برای دریافت بیشترین سود داشته باشیم) به صورت زیر است :

```
( 0 , 3 ) , ( 4 , 6 )
```

به این معنا که در روز صفرم سهم را خریده و سپس در روز سوم آن را بفروشیم . سپس در روز چهارم آن را خریداری کرده و در روز ششم آن را بفروشیم که در نهایت با توجه به مقادیر آرایه حداکثر سود بدست می آید :

```
( StockValues[3] - StockValues[0] ) +
( StockValues[6] - StockValues[4] ) =
( 310 - 100 ) + ( 695 - 40 ) = 865
```

و در بازار دوم یکی از راه های ممکن برای اینکه بیشترین سود را دریافت کنیم (ممکن است در مثالی چندین راه برای دریافت بیشترین سود داشته باشیم) به صورت زیر است :

```
( 3 , 4 )
```

به این معنا که در روز سوم سهم را خریده و سپس در روز چهارم آن را بفروشیم که در نهایت با توجه به مقادیر آرایه حداکثر سود بدست می آید :

```
 StockValues[4] - StockValues[3]  = ( 4 - 2 ) = 2
```

در نتیجه خروجی این تابع ماکسیمم 2 و 865 یعنی 865 را برمیگرداند.

Happy Coding ;)