

## ۴ عنوان

۵ هدف از این تمرین آشنایی با سربارگذاری عملگرها و استفاده از این قابلیت در محاسبات ماتریسی است. همانطور که  
۶ اطلاع دارید، ماتریس، آرایشی منظم از عناصر حقیقی است که در تعدادی سطر کنار هم قرار گرفته‌اند. هریک از این عناصر  
۷ واحد، درایه نام دارد. در این تمرین برای یک ماتریس و سطرهاى آن، مجموعه‌ای از فرایندهای ریاضی را تعریف کرده و  
۸ اپراتورهای متناظر با هر یک را سربارگذاری می‌کنیم. برای رسیدگی به خطاهای احتمالی نیز، از استثناها استفاده خواهیم  
۹ کرد و این مسأله را در تمرین‌های آتی نیز رعایت می‌کنیم. توجه کنید که در این تمرین نیازی به نوشتن تابع main نیست و  
۱۰ فایل‌های شما با استفاده از تابع آماده‌شده توسط دستیاران آموزشی تست خواهد شد.

## ۱۱ کلاس سطر

۱۲ ماتریس از تعدادی سطر هم اندازه که زیر هم قرار گرفته‌اند تشکیل شده است. جدول زیر فرایندهای مرتبط با هر سطر و  
۱۳ عملگر متناظر آنها را مشخص می‌کنند. توجه کنید که متغیر با نام Row نمایانگر یک شیء از نوع سطر و متغیر با نام Matrix  
۱۴ نمایانگر یک شیء از نوع ماتریس است؛ همچنین به جای نام متغیرها در جدول زیر تایپ آن‌ها قرار داده شده است.

Method / Operator	توضیحات
Row(integer)	سطری به طول مشخص شده می‌سازد و تمام درایه‌های آن را برابر با صفر قرار می‌دهد.
Row(vector<double>)	سطری خالی می‌سازد و با بردار مشخص شده مقداردهی می‌کند.
Row1 = Row2	سطر طرف چپ را برابر با سطر طرف راست قرار می‌دهد.*
Row = vector<double>	سطر را برابر با یک بردار از اعداد قرار می‌دهد.*
Row1 + Row2	سطری جدید می‌سازد که درایه‌های آن حاصل جمع درایه‌های متناظر در این دو سطر است.*
Row1 += Row2	درایه‌های سطر سمت راست را به درایه‌های سطر سمت چپ اضافه می‌کند.*
Row1 - Row2	سطری جدید می‌سازد که درایه‌های آن حاصل تفریق درایه‌های متناظر در این دو سطر است.*
Row1 -= Row2	درایه‌های سطر سمت راست را از درایه‌های سطر سمت چپ کم می‌کند.*
Row * Matrix	ضرب ماتریس $1 \times n$ در ماتریس $n \times m$ سطر را در ماتریس ضرب کرده و مقدار حاصل را در سطر جدیدی بازمی‌گرداند.
Row *= Matrix	سطر را در ماتریس ضرب کرده و مقدار حاصل را در سطر قبلی ذخیره می‌کند.
double * Row	یک عدد حقیقی را در درایه‌های سطر ضرب کرده و مقدار حاصل را در سطر جدیدی بازمی‌گرداند.
Row[integer]	رفرنس به درایه n ام سطر را بازمی‌گرداند.
Row[integer] (Row is a constant object)	مشابه عملگر قبل با این تفاوت که روی اشیای ثابت فراخوانی می‌شود.
Row1 == Row2	تساوی دو سطر را بررسی می‌کند.

Row == Matrix	تساوی سطر با ماتریس را بررسی می‌کند. (باید تک‌تک درایه‌های ماتریس و سطر برابر باشند)
integer size(void)	تعداد درایه‌های سطر را بازمی‌گرداند.

۱۵ \* طول دو سطر باید یکسان باشد

## ۱۶ کلاس ماتریس

۱۷ مشابه بخش قبل، جدول زیر فرایندهای مرتبط با هر ماتریس و عملگرهای متناظر آنها را مشخص می‌کنند.

Method / Operator	توضیحات
Matrix()	ماتریسی خالی می‌سازد.
Matrix(integer i, integer c)	ماتریسی شامل ۲ سطر می‌سازد که هر سطر از c درایه صفر تشکیل شده‌است.
Matrix(vector<Row>)	ماتریسی خالی می‌سازد و با برداری از سطرها مقداره‌ی می‌کند.
Matrix(double)	ماتریسی شامل یک سطر می‌سازد و تنها درایه آن سطر را با عدد مشخص شده مقداره‌ی می‌کند.
Matrix1 = Matrix2	ماتریس طرف چپ را برابر با ماتریس طرف راست قرار می‌دهد.
Matrix = vector<Row>	ماتریس را با برداری از سطرها مقداره‌ی می‌کند.
Matrix = Row	ماتریس را برابر با یک سطر قرار می‌دهد.
Matrix = double	ماتریس را برابر با یک عدد حقیقی قرار می‌دهد.
Matrix1 + Matrix2	درایه‌های متناظر دو ماتریس را جمع کرده و مقدار حاصل را در ماتریسی جدیدی بازمی‌گرداند. **
Matrix1 += Matrix2	درایه‌های ماتریس سمت راست را به درایه‌های متناظر در ماتریس سمت چپ اضافه می‌کند. **
Matrix1 - Matrix2	درایه‌های ماتریس سمت راست را از درایه‌های متناظر ماتریس سمت چپ کم کرده و مقدار حاصل را در ماتریسی جدیدی بازمی‌گرداند. **
Matrix1 -= Matrix2	درایه‌های ماتریس سمت راست را از درایه‌های ماتریس سمت چپ کم می‌کند. **
Matrix1 * Matrix2	ماتریس سمت چپ را در ماتریس سمت راست ضرب کرده و مقدار حاصل را در ماتریسی جدیدی بازمی‌گرداند. توجه کنید که تعداد سطرهای ماتریس سمت راست باید با تعداد ستون‌های ماتریس سمت چپ برابر باشد.
Matrix1 *= Matrix2	ماتریس سمت راست را در ماتریس سمت چپ ضرب کرده و مقدار حاصل را در ماتریس سمت چپ قرار می‌دهد.
double * Matrix	یک عدد از نوع double را در تمامی درایه‌های ماتریس ضرب کرده و مقدار حاصل را در ماتریسی جدیدی بازمی‌گرداند.
Matrix1 / Matrix2	ماتریس سمت چپ را در معکوس ماتریس سمت راست ضرب کرده و مقدار حاصل را در ماتریسی جدیدی بازمی‌گرداند. توجه کنید که دو ماتریس باید مربعی و با ابعاد برابر باشند. <sup>۱</sup>
Matrix ^ integer p	یک ماتریس مربعی را به توان یک عدد صحیح می‌رساند. اگر p برابر با صفر بود، ماتریس یک و اگر p کوچکتر از صفر بود، معکوس ماتریس اولیه به توان  p  را بازمی‌گرداند.
-(Matrix)	درایه‌های ماتریس را در -1 ضرب کرده و مقدار حاصل را در ماتریسی جدیدی بازمی‌گرداند.
Matrix[integer]	رفرنس به سطر n ام ماتریس را بازمی‌گرداند.
Matrix [integer] (Matrix is a constant object)	مشابه عملگر قبل با این تفاوت که روی اشیای ثابت فراخوانی می‌شود.

<sup>۱</sup> چگونگی محاسبه ماتریس معکوس با استفاده از روش گاوس جردن ضمیمه شده‌است.

Matrix1 == Matrix2	تساوی دو ماتریس را بررسی می‌کند.
Matrix == Row	تساوی ماتریس با سطر را بررسی می‌کند. (باید تک‌تک درایه‌های ماتریس و سطر برابر باشند)
integer height(void)	تعداد سطرها را باز می‌گرداند.
integer width(void)	تعداد درایه‌های یک سطر را باز می‌گرداند.
double determinant(void)	دترمینان ماتریس را به صورت بازگشتی محاسبه می‌کند. <sup>۲</sup>
void insert(Row, integer)	یک سطر را در اندیس مشخص شده در ماتریس اضافه می‌کند.

۱۸ \*\* ابعاد دو ماتریس باید برابر باشند

## ۱۹ عملگرهای ورودی و خروجی

Method / Operator	توضیحات
1) cin >> Row 2) cin >> Matrix 3) cout << Row 4) cout << Matrix	برای خواندن از ورودی استاندارد و درج در خروجی استاندارد استفاده می‌شوند.

۲۰ (۱) اطلاعات هر سطر در یک خط در ورودی استاندارد وارد می‌شود. درایه‌های یک سطر توسط کاراکتر `\t` از هم جدا می‌شوند.

۲۱ (۲) هر ماتریس با یک براکت شروع شده، در خط بعدی سطر اول، پس از آن سطر دوم و ... به برنامه داده می‌شود؛ پس از آخرین

۲۲ سطر نیز یک براکت در خط بعدی وارد می‌شود. برای مثال:

۲۳  
۲۴  
۲۵  
۲۶  
۲۷

```
[
1      2      3
1.3    7.8    9
12     100    1
]
```

۲۸ (۳) سطر را با همان فرمت قسمت ۱ چاپ کنید. پس از چاپ کردن درایه‌ها، نباید کاراکتر `\n` را در انتهای خط چاپ کنید!

۲۹ (۴) ماتریس را با همان فرمت قسمت ۲ چاپ کنید. پس از چاپ کردن براکت انتهای ماتریس، نباید کاراکتر `\n` را چاپ کنید!

## ۳۰ نکات مهم

- ۳۱ - توجه کنید در محاسبه دترمینان و ماتریس معکوس، اعداد نهایی را با دقت ۳ رقم اعشار چاپ کنید.
- ۳۲ - هشدارهای کامپایلر (warning) را جدی گرفته و سعی کنید آنها را رفع کنید.
- ۳۳ - شما می‌توانید علاوه بر توابع معرفی شده در جداول بالا، توابعی دلخواه جهت افزایش خوانایی و بهبود کدهای خود در نظر بگیرید. اما هنگام تحویل حضوری باید قادر به دفاع از تصمیمات خود باشید!
- ۳۴ - هر نوع استثنائی را که ممکن است هنگام اجرای برنامه رخ دهد، مدیریت کنید و در صورت لزوم کلاس‌های مناسب برای این کار تعریف کنید.
- ۳۵ - به فرمت‌های مطرح شده در صورت تمرین دقت کرده و توابع و عملگرهای خود را دقیقاً مانند جداول بالا تعریف و پیاده‌سازی کنید. در صورت یکسان نبودن فرمت‌ها، تابع `main` نوشته شده برای تست کردن کد شما، با خطای زمان کامپایل روبرو می‌شود و نمره‌ی بخش اجرا را از دست خواهید داد.
- ۳۶
- ۳۷
- ۳۸
- ۳۹

<sup>2</sup> <http://mathworld.wolfram.com/DeterminantExpansionbyMinors.html>

## نحوه‌ی تحویل ۴۰

- ۴۱ فایل‌های مربوط به برنامه‌ی خود را در پوشه‌ای به نام A5-SID.zip را در سایت درس آپلود کنید. (SID پنج رقم آخر شماره‌ی
- ۴۲ دانشجویی شماست. به عنوان مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۴۱۲۳ است، نام فایل شما باید A5-94123.zip باشد.)
- ۴۳ فایل آپلودی شما باید شامل پوشه‌ی کامل پروژه باشد. تحویل این تمرین به صورت حضوری است و در هنگام تحویل باید به تمام
- ۴۴ قسمت‌های کد خود مسلط باشید.

۴۵

## دقت کنید ۴۶

- ۴۷ پروژه‌ی شما باید حتماً **Makefile** داشته باشد. در غیر این صورت نمره بخش اجرا را از دست خواهید داد.
- ۴۸ برنامه‌ی شما باید در سیستم عامل لینوکس نوشته و با کامپایلر g++ کامپایل شود.
- ۴۹ به فرمت و نام فایل‌های خود دقت کنید. در صورتی که هر یک از موارد گفته شده رعایت نشود، نمره‌ی صفر برای بخش اجرای شما در نظر گرفته می‌شود.
- ۵۰
- ۵۱ در صورت کشف تقلب در کل و یا قسمتی از تمرین، برای هر دو طرف نمره‌ی ۱۰۰- منظور خواهد شد.