تمرین اول

کار با ماتریس ها

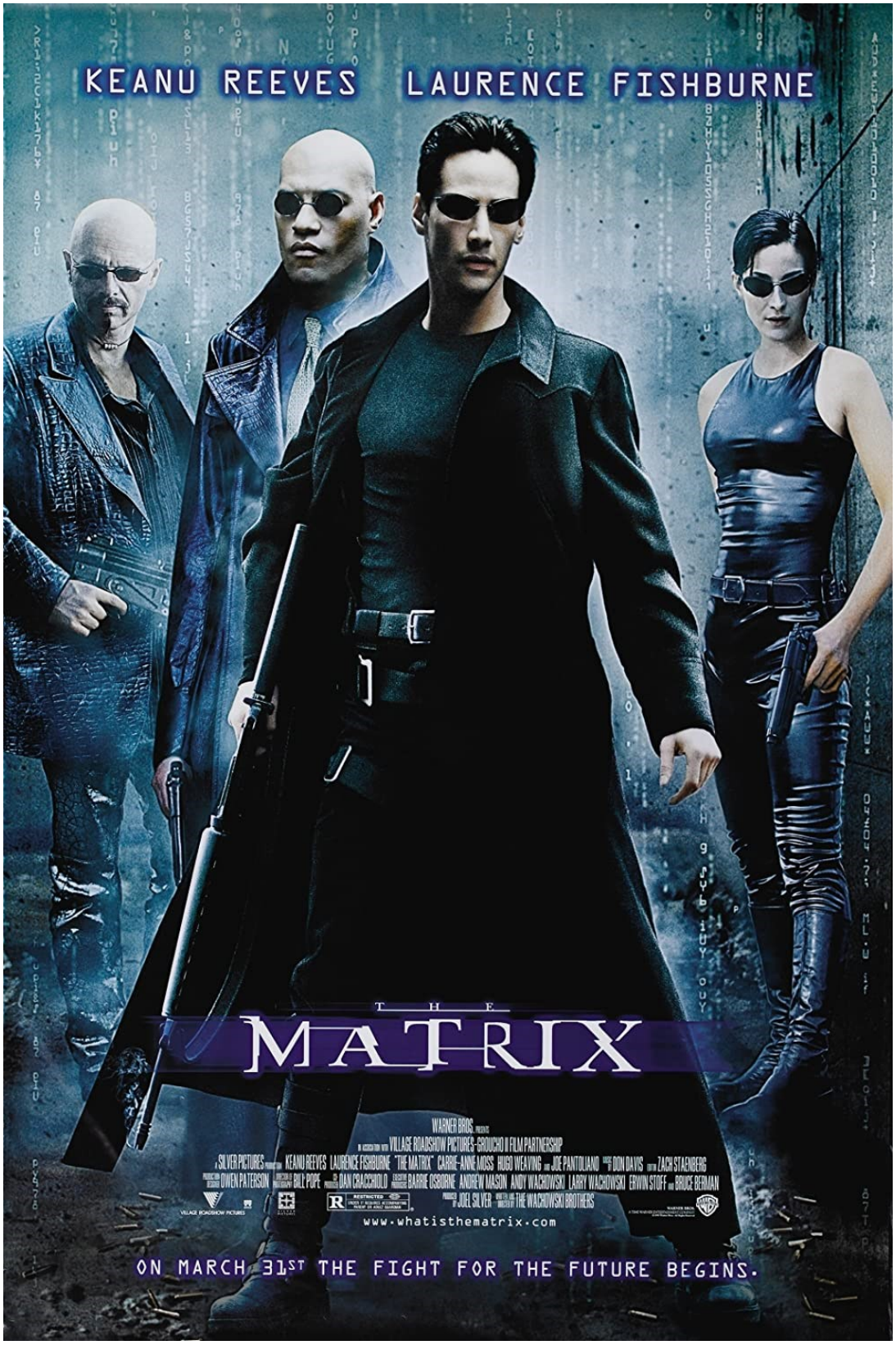
برنامه‌سازی پیشرفته

پاییز 1401

استاد: مهندس نرگس بطحائیان

مهلت تحویل: 20 آذر 1401

زبان c++


KEANU REEVES
LAURENCE FISHBURNE
THE MATRIX
WARNER BROS. PRESENTS
IN ASSOCIATION WITH VILLAGE ROADSHOW PICTURES-GROUCHO II FILM PARTNERSHIP
A SILVER PICTURES PRODUCTION
KEANU REEVES LAURENCE FISHBURNE "THE MATRIX" CARRIE-ANNE MOSS HUGO WEAVING AND JOE PANTOLIANO
MUSIC BY DON DAVIS
EDITED BY ZACH STAENBERG
PRODUCTION DESIGNER OWEN PATERSON
DIRECTOR OF PHOTOGRAPHY BILL POPE
A.S.C.
PRODUCED BY DAN CRACCHIOLO
ASSOCIATE PRODUCERS BARRIE OSBORNE ANDREW MASON ANDY WACHOWSKI LARRY WACHOWSKI ERWIN STOFF AND BRUCE BERMAN
VILLAGE ROADSHOW PICTURES
R
RESTRICTED
PRODUCED BY JOEL SILVER
WRITTEN AND DIRECTED BY THE WACHOWSKI BROTHERS
www.whatisthematrix.com
ON MARCH 31ST THE FIGHT FOR THE FUTURE BEGINS.

ماتریس؟!

ماتریس‌ها یکی از مفاهیم پرکاربرد در دنیای مدرن به شمار میرن. در واقع ماتریس‌ها ساختار های مستطیلی یا مربعی از اعداد، عبارت، نماد، یا داده‌هایی هستن که در تعدادی سطر و ستون چیده شده‌ن و برای کاربردهای متفاوت به کار میرن.

از ماتریس معمولا برای بیان داده ها و یا به عنوان ابزاری مفید برای حل مسائل در زمینه های مختلف علم و زندگی از جمله حل سیستم های معادلات خطی، نظریه گراف، نظریه بازی ها، مدیریت جنگل، ژنتیک، رمزنگاری داده ها، شبکه های الکتریکی و خیلی چیزای باحال دیگه استفاده میشه.

ماتریس یا ماتریکس؟؟

در زبان انگلیسی، به ماتریس matrix میگن، اما توی فرانسوی بهش همون matrice گفته میشه و ما هم ماتریس رو استفاده میکنیم.

مقدمه تمرین

توی این تمرین قراره کار با ماتریس ها رو مرور کنیم و اونهارو وارد دنیای دیجیتال و صفر و یک کنیم. در ادامه چند تا از خاصیت‌های ماتریس‌ها گفته میشه (که یاتون بیاد) و بعدش یه سری تابع و پیاده‌سازی رو مطرح میکنیم که نیازه شما انجام بدید.

برنامه‌ای که قراره توسط شما نوشته بشه باید مثل خط فرمان عمل کنه، یعنی یک منو داره که به کاربر در ابتدای برنامه نمایش داده میشه و یه سری دستور خاص توش نوشته که اگر هرکدوم توسط کاربر وارد شد، اعمال مربوط به اون دستور انجام بشه. مثلا اگر توی خط فرمان کاربر وارد کرد add matrix شما باید مراحل گرفتن یک ماتریس از کاربر رو انجام بدید و ماتریس رو به صورت کامل و صحیح دریافت و ذخیره کنید. همه دستورات و مراحل و حالت‌ها در ادامه گفته میشن.

تیترهای زرد مربوط به تعاریف، و **تیترهای سبز** بخش‌هایی هستن که شما باید پیاده‌سازی کنید.

ماتریس‌های مربعی:

به ماتریسی ـ که تعداد سطر و ستون اون یکی باشه ماتریس مربعی میگیم، یعنی ماتریسی با مربعی 3 تایی به این شکل نمایش داده میشه:

$$Square\ Matrix\ M = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}$$

به همین ترتیب ماتریس های مربعی رو میشه به این شکل هم نمایش داد:

$A_{n*n} = [a_{ij}] \quad | \quad i, j = 1,2,3,4, \ldots$

توی این تمرین ماتریس‌ها رو به شکل دنباله‌ای از اعداد هم نمایش میدیم، جلوتر برای ورودی گرفتن و ذخیره ماتریس‌ها از این روش استفاده میکنیم، مثلا برای ماتریس3 در 2:

$A_{3*2} = [2,5,6,7,8,9]$

که دو عدد اول(2و5) مربوط به سطر اول، دو عدد دوم (6و7) مربوط به سطر دوم، و دو عدد سوم (8و9) مربوط به سطر سوم میشن. توجه کنید فاصله‌ای بین اعداد وجود نداره.

دریافت انواع ماتریس از کاربر:

همونجور که قبلا هم گفته شد، ما انواع ماتریس رو داریم که چند سطر و چند ستون دارن. اولین دستوری که باید پیاده‌سازی کنید دستور add matrix هستش.

نحوه پیاده‌سازی به این صورته که بعد از نام دستور، یک نام به همراه تعدادی عدد وارد میشه که نمایانگر تعداد سطر و ستون هست.

در حالت اول، دو عدد بعد از دستور خواهد اومد. مثلا با دستور زیر، ما آماده دریافت یک ماتریس با 4 سطر و 7 ستون از کاربر میشیم:

```
add matrix fisrt_mat 4 7
```

در حالت دیگه، اگر فقط یک عدد بعد از دستور اومد، یک ماتریس مربعی ساخته میشه:

```
add matrix secodn_mat 5
```

واضحه که بعد از این دستورات، باید خونه‌های ماتریس به ترتیب و دونه دونه از کاربر با پیغام‌های مناسب گرفته بشه. یعنی پیام اول داده میشه، بعد خونه اول گرفته میشه، بعد پیام دوم و بعد خونه دوم و همینجوری...

البته چون دونه‌دونه گرفتن خونه‌های آرایه در دستور بالا برای اعداد بزرگ خیلی خسته کننده میشه، به این فکر افتادیم که یک نوع دیگه از این دستور داشته باشیم، و اونوع دیگه به این صورته:

```
add matrix thrid_mat 2 3 [1,2,3,4,5,22]
```

و همونجور که قبلا گفتیم این نوع نمایش ماتریس به کارمون میاد خیلی راحت و در یک دستور کل ماتریس رو از کاربر میگیریم.

این مدل دستور برای حالت تک عدده هم باید کار کنه، یعنی این حالت:

```
add matrix fuorth_mat 2 [4,6,7,1085]
```

نام‌ها فاصله ندارن، ولی بین کلمه add و matrix و همچنین سایر قسمت های دستور فقط یک فاصله وجود داره.

تضمین میشه همه خونه‌های ماتریس از یک نوع هستن، یعنی همه یا int یا float یا char یا string و ....

در حالت‌های ناصحیح دیگه، باید پیغام‌های خطای مناسبی به کاربر نمایش داده بشه.

ماتریس قطری:

اگر همه خونه‌های خارج از قطر یک ماتریس صفر باشن، ماتریس قطری در اختیار داریم. معلومه که ماتریسی میتونه قطری باشه که مربعی باشه.

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

قطری است؟؟

یکی از کارهایی که بعد از دریافت و ذخیره کامل ماتریس، باید کاربر قادر باشه انجام بده، چک کردن این موضوعه که آیا ماتریسی که وارد کرده قطری هست یا نه؟

برای اینکار از دستور زیر استفاده میکنه:

```
is_diagonal fisrt_mat
```

اگر قطری بود میگیم آره و اگر نبود نه گفته میشه

ماتریسی که همه اعضای اون 0 باشه هم قطری حساب میشه.

ماتریس بالا مثلثی و پایین مثلثی:

ماتریسی که همه اعضای بالایی قطر اون صفر باشه: پایین مثلثی؛

و ماتریسی که همه اعضای پایینی قطر اون صفر باشه، ماتریس بالا مثلثی گفته میشه.

شکل زیر مثالی از ماتریس پایین مثلثی هست:

$$\begin{matrix} 1 & 0 & 0 \\ 4 & 2 & 0 \\ 6 & 5 & 1 \end{matrix}$$

بالا یا پایین؟

احتمالا میتونید این رو حدس بزنید... آفرین این پیاده سازی برای تشخیص بالا یا پایین مثلثی بودن ماتریسه. دستورات مورد نیاز برای این بخش به صورت زیره:

```
is_upper_triangular second_mat

is_lower_triangular second_mat
```

برای این دو نوع دستور، اره یا نه برگردونده میشه، ولی برای حالت جامع یعنی دستور زیر باید گفته بشه که بالا مثلثی هست یا پایین؟ یا شایدم هردو؟!

```
is_triangular third_mat
```

ماتریس همانی:

اگر یک ماتریس قطری باشه و همه اعضای قطر اون 1 باشن، ماتریس همانی بهش گفته میشه.

همونی یا همینی؟

این هم شبیه همون قبلی‌هاست، باید با دستور مناسب، چک کنید که آیا ماتریس همانی هست یا نه. دستور به شکل زیره:

```
is_identity fisrt_mat
```

ماتریس متقارن:

اگر جای سطر و ستون یک ماتریس رو با هم عوض کنیم و به همون ماتریس قبلی برسیم، میگیم ما یه ماتریس متقارن در اختیارمون هست. ماتریس زیر رو داشته باشید:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & -5 \\ 3 & -5 & 6 \end{bmatrix}.$$

ماتریس پادمتقارن:

این ماتریس شبیه همین قبلیه، فقط یه تفاوت کوچیک که داره اینکه انگار در یک منفی هم ضرب شده. مثال زیر رو در نظر داشته باشید، بالایی ماتریس اولیه و پایینی ماتریس پادمتقارن نسبت به بالایی هست:

$$A = \begin{bmatrix} 0 & 3 & -2 \\ -3 & 0 & 4 \\ 2 & -4 & 0 \end{bmatrix}$$

$$\Rightarrow A^T = \begin{bmatrix} 0 & -3 & 2 \\ 3 & 0 & -4 \\ -2 & 4 & 0 \end{bmatrix} = -A$$

پادمتقارن یا بدون-پادمتقارن:

توی این بخش باید چک کنید ماتریس متقارن هست یا نه، و دستور دیگه هم پادمتقارن بودن یا نبودن. در زیر مشاهده میکنید:

```
is_normal_symmetric sixht_mat

is_skew_symmetric sixht_mat
```

شبیه همون قبلی ها، برای این مورد هم یک دستور جامع داریم که باید مشخص کنه متقارن هست یا نه، و اگر هست چه نوع تقارنی داره:

```
is_symmetric sixth_mat
```

معکوس:

در تعاریف اصلی ماتریس ها، برای معکوس کردن یک ماتریس، باید جای چند خونه با هم عوض بشه و بعضی‌ها ثابت بمونن، بعد بعضی‌ها در منفی ضرب بشن و بعد همه خونه‌ها تقسیم بر دترمینان ماتریس بشن. ما اینجا از یک تعریف دیگه برای معکوس کردن استفاده میکنیم، فقط کافیه جای سطر و ستون عوض بشه. همین. در واقع چیزی که توی بخش متقارن بودن چک میکردید رو اینجا باید ذخیره کنید.

معکوسینگ:

با دستور زیر، ماتریس معکوس شده و ذخیره میشه.

اگر نامی در ادامه دستور داده شد، ماتریس فعلی دست نخورده نگه‌داشته میشه و معکوس آن به عنوان یک ماتریس جدید ذخیره میشه. اگر نامی آورده نشده بود همون ماتریس تبدیل به معکوس خودش میشه. به این صورت‌ها:

```
inverse mat_1

inverse mat_2 mat_5
```

نمایش ماتریس:

باید با گرفتن نام ماتریس بعد از کلمه show اون رو به صورت مناسب چاپ کنید.

حذف ماتریس:

کاربر باید بتونه با دستور delete و بعد از اون نام ماتریس، اون رو به کلی حذف کنه.

ویرایش یک خانه:

برای ویرایش یک خونه از ماتریس دستور زیر اجرا میشه:

```
change mat_name 4 3 1563
```

اول نام ماتریس ذکر میشه، بعد شماره سطر و بعد شماره ستون، در آخر هم مقدار جدید.

نکات قابل توجه:

- استفاده از کامنت و تهیه گزارش کار **اجباریه**، بخش زیادی از نمره به کامنت و نحوه گزارش نویسی مناسب اختصاص داره. البته این هم بدونید اینکه فقط یه خط کامنت یا گزارش نوشته باشید و رفع مسئولیت کرده باشید مورد قبول نیست و بخش زیادی از نمره رو از دست خواهید داد. کامنت ها باید معنا دار و اصولی باشن، و گزارش کار باید درست و اصولی و همچنین زیبا باشه.
- استفاده از گیت و بیلد سیستم، شبیه cmake **اجباریه** و بخش زیادی از نمره رو به خودش اختصاص میده.
- ساختار برنامه‌ای که خواهید نوشت باید به این صورت باشه که یک تابع main دارید، در اون یک تابع برای تشخیص نوع دستور فراخوانی میکنید، و بعد از اینکه دستور رو تشخیص دادید تابع مخصوص به اون حالت دستور رو فراخوانی میکنید.
- راه‌حل‌های ابداعی خودتون در صورتی که جواب درست بدن و جالب باشن نمره مثبت دارن.
- استفاده از توابع بازگشتی برای حل سوالات، نمره مثبت اضافه دارن.
- برای نوشتن برنامه از تابع استفاده کنید، برنامه باید **ماژولار** بوده و تا حد امکان تابع main خلوت باشه.
- از متغیرها با اسامی با معنا استفاده کنید.
- واضحه که برای حالات استثنا و خطا، باید ارور مناسب نمایش داده و فعالیت درست و مناسب انجام شود.

- استفاده از function overloading مجاز **نیست**،باید از default argument ها برای تمیز حالات استفاده کنید.
- برای کسانی که متذکر بشن اسم ماتریس ها در نمونه‌ها اشتباه تایپی داره نمره منفی اعمال خواهد شد. :)
- اگر تشخیص دستورات و کار با استرینگ‌ها براتون سخته، میتونید یه منو تهیه کنید و برای هر آیتم شماره بذارید، و از کاربر به جای دستور، شماره‌ی اون دستور رو بگیرید. از حالت قبلی راحت تره و سریع تر پیاده سازی میشه. باید توجه داشته باشید که نحوه دریافت اطلاعات هم در این روش باید کامل و شبیه حالت قبلی باشه. البته اگر با این روش نوشتید، 35% نمره رو از دست میدید و نمره شما از 65 محاسبه میشه، ولی اگر اون خیلی سختتونه اینجوری بنویسید.

موفق باشید..!