


مبانی برنامه سازی - پاییز ۹۹ پروژه درسی	مدرس:ان: مهدی زمانیان سید بهروز نصیحت کن	 <div>                     دانشگاه صنعتی خواجه نصیرالدین طوسی                      K. N. TOOSI UNIVERSITY OF TECHNOLOGY                 </div>
--	--	---

## برنامه محاسبات ماتریسی

می خواهیم برنامه ای بنویسیم که محاسبات ماتریسی معمول را برای ما انجام دهد. انتظار داریم که این برنامه بتواند عملیات:

- دریافت ماتریس از کاربر
- نمایش ماتریس
- جمع دو ماتریس
- تفریق دو ماتریس
- ضرب دو ماتریس
- ترانواده یک ماتریس
- دترمینان یک ماتریس
- معکوس یک ماتریس
- خواندن محتوای یک ماتریس از فایل
- نوشتن محتوای یک ماتریس در فایل

را برای ما انجام دهد. به این منظور در ابتدا ساختاری به نام **Matrix** تعریف کنیم که اطلاعات ماتریس در آن قرار داده می شود. این اطلاعات شامل آرایه ای دو بعدی از نوع اعشاری برای نگهداری اعضای ماتریس و دو عدد صحیح  $m$  و  $n$  برای نگهداری تعداد سطر و ستون ماتریس است.

```
#define MAX_SIZE 20

struct Matrix {
    float array[MAX_SIZE][MAX_SIZE];
    int m,n;
};
```

برای انجام عملیات ذکر شده در بالا نیاز به پیاده سازی تعدادی تابع داریم که چند تا از آنها در زیر معرفی شده اند

- تابع **readMatrix** ابعاد و درایه های یک ماتریس را از کاربر دریافت می کند و یک متغیر از نوع **Matrix** بر می گرداند.
- تابع **printMatrix** ابعاد و درایه های ماتریس ورودی تابع را نمایش می دهد.
- تابع **addMatrix** دو ماتریس ورودی از نوع اشاره گر به ساختار **Matrix** را جمع کرده و حاصل را در اشاره گر ورودی سوم قرار می دهد.
- تابع **subMatrix** دو ماتریس ورودی از نوع اشاره گر به ساختار **Matrix** را از هم کم کرده و حاصل را در اشاره گر ورودی سوم قرار می دهد.
- تابع **mulMatrix** دو ماتریس ورودی از نوع اشاره گر به ساختار **Matrix** را در هم ضرب کرده و حاصل را در اشاره گر ورودی سوم قرار می دهد.
- سه تابع بالا یک مقدار بازگشتی از نوع **int** دارند. در صورتی که ابعاد دو ماتریس اول برای انجام عملیات مورد نظر با هم سازگار باشند تابع عملیات را انجام می دهد و به عنوان مقدار بازگشتی عدد 1 را بر می گرداند. در صورتی که ابعاد ماتریس ها سازگار نباشند عملیات انجام نمی شود و تابع مقدار 0 را بر می گرداند.

الگوی این توابع به صورت زیر است:

```
struct Matrix readMatrix();  
void readMatrix(struct Matrix a);  
  
int addMatrix(struct Matrix *a, struct Matrix *b, struct Matrix *c);  
int subMatrix(struct Matrix *a, struct Matrix *b, struct Matrix *c);  
int mulMatrix(struct Matrix *a, struct Matrix *b, struct Matrix *c);
```

برای پیاده سازی این تابع نیاز است که تعدادی تابع که بخشی از آنها معرفی شد و الگوی بخشی دیگر را خودتان باید بنویسید و همه تابع ها را پیاده سازی کنید.

در ابتدای شروع برنامه منویی به کاربر نشان دهید و عملیات مورد نظر کاربر را از او دریافت نمایید. بسته به عملیات انتخاب شده ممکن است نیاز به دریافت ماتریسها از کاربر و یا از فایل باشد. پس از دریافت اطلاعات پردازشهای مورد نیاز را انجام داده و به کاربر نشان دهید. در صورت درخواست کاربر لازم است نتایج بر روی فایل هم ذخیره شوند. حداقل قابلیت هایی که منو باید داشته باشید از نوع زیر است.

- دریافت ماتریس اول A از ورودی
  - دریافت ماتریس دوم B از ورودی
  - دریافت ماتریس اول A از فایل
  - دریافت ماتریس دوم B از فایل
  - نشان دادن دترمینان ماتریس A
  - انجام عملیات ضرب، جمع یا تقریق روی دو ماتریس A و B و ذخیره حاصل در ماتریس A
  - انجام عملیات ترانهاد، یا معکوس روی ماتریس A و ذخیره آن در ماتریس A
  - ذخیره کردن ماتریس A در یک فایل.
- می توانید منوی پیشرفته تری نیز بسازید.

منابعی برای برنامه نویسی فایلها که می توانید آنها را مطالعه کنید:

[https://www.tutorialspoint.com/cprogramming/c\\_file\\_io.htm](https://www.tutorialspoint.com/cprogramming/c_file_io.htm)

<http://www.cprogramming.com/tutorial/cfileio.html>

<https://www.codingunit.com/c-tutorial-file-io-using-text-files>

<http://beginnersbook.com/2014/01/c-file-io>

برای کار با فایل همچنین می توانید به ویدیوی درس سال پیش مراجعه نمایید.

<https://www.aparat.com/v/UskvG?playlist=258064>

برای بدست آوردن دترمینان معکوس یک ماتریس می توانید از کهاد (minor) ها به صورت بازگشتی استفاده کنید. هرچند الگوریتم های به مراتب سریع تری هم وجود دارند.

<https://fa.wikipedia.org/wiki/%DA%A9%D9%87%D8%A7%D8%AF>

[https://en.wikipedia.org/wiki/Minor\\_\(linear\\_algebra\)](https://en.wikipedia.org/wiki/Minor_(linear_algebra))

## توجه

- پروژه ها باید به صورت تک نفره انجام شوند. می توانید با دیگران مشورت کنید، ولی به هیچ عنوان قطعه ای از کد خود را در اختیار فرد دیگری قرار ندهید.
- موارد خواسته شده را تا جایی که می توانید انجام دهید. به نسبت مواردی که پیاده سازی شده نمره دریافت خواهید کرد.