

## گزارش کار تمرین شماره ۱ درس برنامه نویسی پیشرفته

```
Matrix zeros(size_t n, size_t m)
```

-در این تابع ابتدا یک بردار به تعداد ستون های ماتریس تشکیل داده شده سپس هر یک از مقادیر بردار با استفاده از دستور `resize` به اندازه ی تعداد ستون ها گسترش پیدا کردند.

```
Matrix ones(size_t n, size_t m)
```

-در این تابع مانند تابع `zeros` تعداد سطرها و ستون ها مشخص شدند و بعد هریک از درایه های آن را با عدد ۱ مقداردهی شدند.

```
Matrix random(size_t n, size_t m, double min, double max)
```

```
void show(const Matrix& matrix)
```

-با استفاده از دستور `decimal place. setprecision` را برابر با ۳ قرار دادیم.

```
Matrix multiply(const Matrix& matrix, double c)
```

-با استفاده از دو حلقه ی تو در تو تمام درایه های ماتریس را در متغیر `C` ضرب می کنیم.

```
Matrix multiply(const Matrix& matrix¹, const Matrix& matrix²)
```

-با استفاده از سه حلقه ی تو در تو (حرکت در سطرها ی ماتریس اول-حرکت در ستون های ماتریس دوم-حرکت بین درایه های سطرها ی ماتریس اول و ستون های ماتریس دوم) ضرب ماتریسی انجام شده است.

```
-Matrix sum(const Matrix& matrix, double c)
```

--با استفاده از دو حلقه ی تو در تو تمام درایه های ماتریس را با متغیر `C` جمع کردیم.

```
Matrix sum(const Matrix& matrix¹, const Matrix& matrix²)
```

-با استفاده از دو حلقه ی تو در تو تمامی درایه های متناظر ماتریس با یکدیگر جمع می شوند.

```
Matrix transpose(const Matrix& matrix)
```

-با استفاده از دو حلقه ی تو در تو هر درایه را با درایه ای با سطر و ستون جابجا متناظر می کنیم.

```
Matrix minor(const Matrix& matrix, size_t n, size_t m)
```

-ابتدا متغیر خروجی به اندازه ی یک واحد کمتر در ستون و یک واحد کمتر در ردیف تعریف می کنیم و دو متغیر برای `index` ماتریس خروجی تعریف می کنیم و مقدار اولیه آنها را برابر با یک قرار می دهیم. ابتدا در حلقه ی اول که مربوط به ستون هاست چک می کنیم که آیا به ستون مرد نظر رسیده ایم یا خیر اگر رسیده باشیم با دستور `continue` مجدد به حلقه باز می کردیم در غیر این صورت یک واحد به `index` ردیف های ماتریس خروجی اضافه می کنیم و وارد حلقه ی دوم می شویم و همان شرط برای

ردیف ها را در ستون ها پیاده می کنیم و index ستون های ماتریس خروجی را تغییر می دهیم و در نهایت اگر وارد هیچ یک از دو شرط نشود ماتریس خروجی را با در اختیار داشتن index ها مقداردهی می کنیم.

```
double determinant(const Matrix& matrix)
```

-با استفاده داده های ردیف صفر ماتریس و محاسبه ی دترمینان ماتریس ماینور (دترمینان را برای ماتریس دو در دو با استفاده از رابطه داده شده و برای ماتریس هایی با مرتبه ی بالاتر با خود فراخوانی تابع محاسبه می شود).

```
Matrix inverse(const Matrix& matrix)
```

-ابتدا دترمینان ماتریس ماینور را برای تک تک درایه ها محاسبه می کنیم و در  $(-1)^{(i+j)}$  ضرب می کنیم و ماتریس حاصل را ترانپوذه می کنیم و تمامی درایه ها را در معکوس دترمینان ضرب می کنیم.

```
Matrix concatenate(const Matrix& matrix¹, const Matrix& matrix²,int axis)
```

- ماتریس ۱ را درایه به درایه به ماتریس خروجی منتقل می کنیم و ماتریس دوم را با دو حلقه ی تو در تو کمی متفاوت تر در ماتریس خروجی میریزیم.

اگر axis=۰ باشد:

```
for(size_t i=n¹;n¹+n²;i++)
    for(size_t j=۰;j<m²;j++)
        matrix¹_co[i][j]=matrix²[i-n¹][j];
```

اگر axis=۱ باشد:

```
for(size_t i=۰;i<n²;i++)
    for(size_t j=m¹;j<m²+m¹;j++)
        matrix¹_co[i][j]=matrix²[i][j-m¹];
    }
}
```

```
Matrix ero_swap(const Matrix& matrix, size_t r¹, size_t r²)
```

-با استفاده از یک متغیر واسط درایه هاس ستون های ۲۱ و ۲۲ را با یکدیگر جابجا می کنیم.

```
Matrix ero_multiply(const Matrix& matrix, size_t r, double c)
```

-با استفاده از یک حلقه تمامی درایه های ردیف ۲ را در متغیر C ضرب می کنیم.

```
Matrix ero_sum(const Matrix& matrix, size_t r¹, double c, size_t r²)
```

-با استفاده از یک حلقه تمامی درایه های ردیف ۲۱ را در C ضرب می کنیم و با درایه های ردیف ۲۲ جمع می کنیم و درایه های خروجی را در همان ماتریس ۲۲ ذخیره می کنیم.

```
Matrix upper_triangular(const Matrix& matrix)
```

-ابتدا با بررسی می کنیم درایه مربوطه در سطر مورد نظر(ردیف صفر و ستون صفر،ردیف یک و ستون یک و...)برابر با صفر است یا خیر اگر برابر با صفر بود با ردیف های پایین تر که غیر صفر است جابجا می کنیم سپس با استفاده از یک حلقه و دستور ero\_sum

ردیف مورد نظر را با ضریبی که بتواند درایه های هم ستون که در ردیف های پایین تر قرار دارند را صفر کند ضرب کرده و با ردیف های پایین تر جمع می کنیم.