

## عملیات ماتریسی

در این سوال می خواهیم یک ماشین حساب برای انجام محاسبات روی ماتریس ها طراحی کنیم که عملیات زیر را پوشش میدهد :

- جمع دو ماتریس
- تفریق دو ماتریس
- ضرب دو ماتریس
- چک برابری دو ماتریس
- محاسبه دترمینان یک ماتریس
- محاسبه معکوس یک ماتریس

شما باید برای تک تک عملیات مدنظر تابع هایی با امضاهای زیر پیاده سازی کنید. اسمی توابع و پارامتر ها اختیاری بوده ولی نوع و تعداد پارامتر ها باید همین شکلی باشند

**جمع:** `void add(matrix1, matrix2, matrix3, row, column)`

این تابع دو ماتریس با ابعاد برابر ورودی گرفته و نتیجه را در ماتریس سوم ذخیره میکند

**تفریق:** `void sub(matrix1, matrix2, matrix3, row, column)`

این تابع حاصل  $\text{matrix1} - \text{matrix2}$  را در `matrix3` ذخیره میکند

**ضرب :** `void mul(matrix1, matrix2, matrix3, row1, column1, column2)`

این تابع ماتریس 1 با ابعاد `row1, column1` را در ماتریس 2 با ابعاد `column1, column2` ضرب کده و نتیجه را در ماتریس 3 ذخیره میکند

**چک برابری :** `bool is_equal(matrix1, matrix2, row1, column)`

این تابع دو ماتریس با ابعاد برابر ورودی گرفته و چک میکند که دو ماتریس برابر هستند یا خیر

`long double determinan(matrix1, row, column)` دترمینان:

این تابع دترمینان ماتریس مربعی با ابعاد `row`, `column` را حساب و نتیجه را خروجی میدهد از بسط لایپلاس (بسط بر اساس یک سطر یا ستون) برای محاسبه دترمینان استفاده کنید دقت کنید که باید این تابع را به صورت بازگشتی پیاده سازی کنید

`bool inverse(matrix1, matrix2, row, column)` : وارون ماتریس :

این تابع وارون ماتریس مربعی 1 را حساب و در ماتریس مربعی 2 ذخیره میکند . اگر وارون سازی موفقیت امیز بود  
برمیگرداند ) `false` صورت این در `غير` و `ture` برگردانده منبع برای آموزش نحوه محاسبه وارون ماتریس های مرتبه بالاتر

`void input(matrix, row, column)` ورودی گرفتن ماتریس

`void print(matrix, row, column)` چاپ ماتریس

نکته مهم :

دقت کنید هیچ کدام از توابع فوق ماتریس اولیه را تغییر نمی دهند

تابع `int main`

بعد نوشتن این تابع داخل تابع `main` ابتدا یک ماتریس با ابعاد `m`, `n` ورودی بگیرید و بعد با نوشتن یک حلقه منوی ایجاد کنید تا کاربر بتواند به دلخواه خود از توابع فوق استفاده کند . گزینه های منو :

- `sub` : در این گزینه یک ماتریس دیگر ورودی گرفته شده و با تابع تفریق ، ماتریس ثانویه را از اولیه کم کرده و در نهایت حاصل را با تابع چاپ ماتریس ، چاپ می کند
- `add` : در این گزینه یک ماتریس دیگر ورودی گرفته شده و با تابع جمع ، جمع دو ماتریس را حساب کرده و در نهایت حاصل را چاپ میکند
- `inverse` , `determinan` , `equal` , `mul` نیز مشابه گزینه های بالایی باید پیاده سازی شوند
- `exit` با وارد شدن این گزینه برنامه تمام می شود

توجه کنید در این منو باید چک کنید که ابعاد ماتریس ها برای انجام عملیات مناسب باشند . در صورت مناسب نبودن ابعاد ، عبارت `size` را چاپ کنید . در محاسبه واورن ماتریس اگر دترمینان صفر بود داخل تابع `determinan is zero!!` عبارت `inverse` را چاپ کنید

## وروڈی

در خط اول ابتدا تعداد سطر ماتریس ( $m$ ) و بعد تعداد ستون ماتریس ( $n$ ) ورودی داده می شود در  $m$  خط بعدی عناصر ماتریس ورودی داده میشوند

در خطوط بعدی ابتدا درخواست کاربر وارد شده و سپس در صورت نیاز گزینه انتخاب شده ماتریس دوم به شکل ماتریس اول ورودی داده می شود

## خروجی

بعد از انجام هر عملیات مدنظر کاربر ، در صورتی که خروجی ماتریس است ، ماتریس را چاپ کنید و در صورتی که خروجی عدد یا عبارت است آن را چاپ کنید چاپ خطاهای فراموشتوان نشه !

## وروڈی نمونه 1

```
2 2
1 2
3 4
determinan
inverse
add
2 2
1 1
1 1
sub
2 2
1 1
1 1
exit
```

## خروجی نمونه 1

-2  
-2 1  
1.5 -0.5  
2 3  
4 5  
0 1  
2 3