هدف از این پروژه، نوشتن یک ماشین حساب ماتریسی ساده است. نحوه کارکرد این ماشین حساب به این صورت است که کاربر قبل از اینکه بتواند عملیاتی را بر روی یک ماتریس انجام دهد، در ابتدا باید آن ماتریس را تعریف کند. بعد از اینکه ماتریس تعریف شد، می‌توان از آن برای انجام محاسبات استفاده کرد. محاسباتی که این ماشین حساب می‌تواند انجام دهد شامل چند عملگر اصلی و چند تابع است. هر دستور محاسباتی می‌تواند بیشتر از یک عملگر یا تابع را استفاده کند. وقتی که این ماشین حساب اجرا شد منتظر دستور از سمت کاربر می‌ماند. بعد از اینکه کاربر دستور را وارد کرد آنرا اجرا می‌کند نتیجه را نشان می‌دهد و دوباره منتظر دستور می‌ماند، زمانی که منتظر دستور است در خط فرمان علامتی مانند علامت < را نشان می‌دهد. در ادامه جزییات این ماشین حساب شرح داده شده است.

۱- دستورات

۱-۱- دستور def: از این دستور برای تعریف یک ماتریس جدید استفاده می‌شود. برای تعریف یک ماتریس جدید دو روش وجود دارد. روش اول، تعریف با استفاده از اعضای آرایه. برای مثال

```
>
> def A=[2,2,2;3,3,3]
    A:
        2  2  2
        3  3  3
>
```

یک ماتریس 2x3 را تعریف کرده است. بعد از اجرای دستور def مقدار ماتریس توسط برنامه نشان داده شده است.

روش دوم، تعریف با استفاده از عملیات ریاضی. برای مثال فرض کنید ماتریس A و B هر دو ماتریس‌های 2x3 و همه اعضای آنها صفر است، آنگاه

```
>
> def C = A + B
    C:
        0  0  0
        0  0  0
>
```

اگر ماتریسی قبلا به این اسم تعریف شده باشد، تعریف جدید جایگزین تعریف قبلی می‌شود (تعریف قبلی از بین می‌رود).

۱-۲- دستور show: این دستور برای نشان دادن همه متغیرهای تعریف شده در ماشین حساب بکار می‌رود. برای مثال اگر A و B دو ماتریس تعریف شده باشند که مقدار همه اعضای A برابر ۱ و اعضای B برابر ۲ باشد در این صورت

```
>
> show
    A:
       1  1  1
```

```
      1   1   1

    B:
      2   2   2
      2   2   2
      2   2   2
>
```

۱- ۳- دستور clear: همه متغیرهای تعریف شده را از بین می‌برد.

```
>
> clear
> show
>
>
```

۱-۴- دستور exit: با این دستور برنامه ماشین حساب تمام می‌شود.

۲- عملیات‌های اصلی

۱-۲- *: دو ماتریس را ضرب می‌کند

۲-۲- + : دو ماتریس را باهم جمع می‌کند.

۳-۲- – : دو ماتریس را از هم کم می‌کند.

۴-۲- @: اعضای ماتریس‌ها را نظیر به نظیر ضرب می‌کند.

۵-۲- / : اعضای ماتریس‌ها را نظیر به نظیر تقسیم می‌کند.

۶-۲- ^: اعضای ماتریس را نظیر به نظیر به توان می‌رساند.

برای مثال:

```
>
> def A = [2,2; 3, 3]

A:
     2     2
     3     3

> def B = [4,4;5,5]

B:
     4     4
     5     5

> A * B

    18    18
    27    27
```

```
> A - B

    -2    -2
    -2    -2

> A + B

     6     6
     8     8

> A @ B

     8     8
    15    15

> A / B

    0.5000    0.5000
    0.6000    0.6000

> A ^ B

    16    16
   243   243
```

۳- توابع

۳-۱- ترانهاده: trans(A)

۳-۲- trace(A): حاصل جمع اعضای قطر اصلی

۳-۳- دترمینان: det(A)

۳-۴- معکوس: invert(A)

۳-۵- تست متقارن بودن: sym(A) اگر متقارن باشد برابر ۱ در غیر این صورت صفر

۴- نکات

مواردی را که گفته شده است "به برنامه داده نمی‌شود"، به این معنی است که برنامه شما لازم نیست این موارد را بررسی کند و پیغام خطا چاپ کند.

۴-۱- فرض بر این است کاربر دستورات را درست وارد می‌کند. برای مثال دستورات زیر هیچ گاه به برنامه داده نمی‌شود:

```
>
> def A = [2, 2; 3, 3, 3]
> def A = [2, 2, X, Y; 3, 3, 3]
> def A = [2, 2; 3, 3), 3
>
```

۴-۲- فرض براین است که اگر بیشتر از یک عملگر در عملیات باشد، همه آنها دارای الویت یکسان هستند. برای مثال دستورات زیر به برنامه داده نمی‌شود:

```
> A = B + C * D
> A = B / C + D ^ E
```

اما دستورات زیر دستورات معتبری هستند که باید برنامه برای آنها درست کار کند

```
> A = B + C – D + E
> A = B * C * D
> A = A @ B @ C
```

الویت‌های دستورات به صورت زیر است:

عملگر	الویت
*	1
+ -	2
^	3
/ @	4

اگر بیش از یک عملگر وجود داشته باشد محاسبه از چپ به راست انجام می‌شود.

۴-۳- تنها حالت‌هایی که کاربر ممکن است اشتباه کند: ۱) اندازه ماتریس‌ها با عملیات مورد نظر همخوانی نداشته باشد برای مثال عملیات ^ برروی دو ماتریس با ابعاد مختلف تعریف نشده است. ۲) از یک متغیر تعریف نشده استفاده کند. در این حالت‌ها برنامه باید پیغام خطا ایجاد کند.

۴-۴- اسم متغیرها تک حرفی است.

۴-۵- امکان استفاده از توابع در عملیات ریاضی وجود دارد. برای مثال دستور زیر یک دستور درست است:

```
> A = invert(A) + trans(B)
```

۴-۶- امکان استفاده از پرانتز به غیر از توابع وجود ندارد.

۴-۷- امکان نوشتن تابع‌های تو در تو وجود ندارد. دستور زیر هیچگاه به برنامه داده نمی‌شود:

```
> A = invert(trans(B))
```

۸-۴- جاهای خالی هیچ تاثیری در عملیات ریاضی ندارند. هر دو عملیات زیر معادل است

```
> A=B+C*D
> A  =  B    + C *D
```

۵- **راهنمایی**:

۱-۵- همه ورودی و خروجی‌ها را به صورت رشته از کاربر بگیرید.

۲-۵- برای هر ماتریسی علاوه بر خود ماتریس، موارد دیگری مانند اسم، اندازه و ... را باید ذخیره کنید که می‌توانید از struct استفاده کنید.

۳-۵- یکی از مهمترین بخش‌های این برنامه پردازش رشته‌ای است که از ورودی گرفته شده است. با توجه به اینکه فرض شده است کاربر ورودی خطادار وارد نمی‌کند، یک رشته ورودی شامل ۱-دستور ۲- اسم متغیر ۳- تابع و ۴- عملگرها است. برنامه باید به درستی هر قسمت رشته را جدا کند و بر اساس آنها تصمیم بگیرد.