

**تمرین شماره 1 برنامه نویسی پیشرفته**

**کار با ماتریس**

**استاد: مهندس نرگس السادات بطحائیان**

**محمدرضا حیدرنیا – 9912358018**

پاییز 1400

**شرح پروژه**

با توجه به متن تمرین باید برنامه ایی می نوشتیم که بتوان از طریق برنامه به راحتی با ماتریس ها کار کرد. در واقع ما دستوری رو که توسط کاربر وارد می شد تحلیل میکردیم و دستور رو اجرا میکردیم.

این تمرین یک چالش اصلی داشت که چگونه نوع داده ایی را مشخص کنیم ، البته تجزیه و تحلیل دستور ورودی هم یک چالش جدی بود.

من در این برنامه چهار نوع داده رو پیاده سازی کردم:

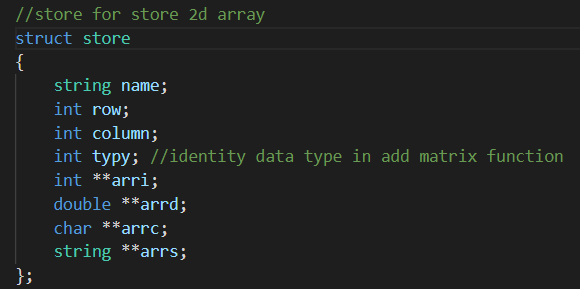
Int – double – string – char

و یک استراکت برای ذخیره آرایه های دوبعدی و سطر و ستون و ....

در این برنامه قبل از هر پردازشی باید مشخص میشد که ماتریس چه نوع داده ای است که یکی از دلایل زیاد شدن بدنه های توابع بود.

**ساختار برنامه**

در این برنامه دو فایل main.cpp و function.h که تمامی توابع در function.h پیاده سازی شده اند، وجود دارد. با استفاده از struct زیر:

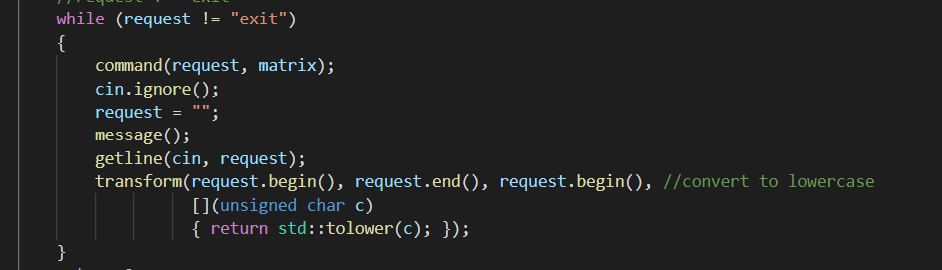


آرایه ها را ذخیره میکنیم. متغیر type برای ذخیره کردن اعداد 1 تا 4 کاربرد دارد که 1 نشان دهنده int ، 2 نشان دهنده double ، 3 نشان دهنده char ، 4 نشان دهنده string.

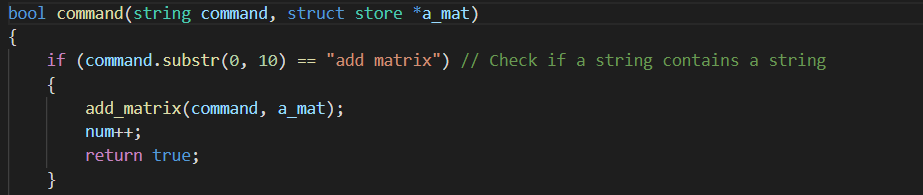
این استراکت در تمامی توابع استفاده شده است.

در ابتدای تابع main تابع message نمایش داده میشود تا تمامی توانایی های برنامه به کاربر نمایش داده شود سپس با استفاده از getline رشته ایی را بدون حساسیت به فاصله دریافت میکنیم البته برای احتیاط دستور گرفته شده را به حروف کوچک تبدیل میکنیم.

در این تابع با استفاده از حلقه دائما منتظر دستور هستیم تا زمانی که کاربر کلمه exit را تایپ کند سپس از حلقه خارج میشویم. دستور گرفته شده به تابع commandفرستاده میشود تا تحلیل شود.



در تابع command با استفاده از substr و طول مشخص، دستور را برسی میکنیم که اگر دستور وارد شده اشتباه بود برنامه اخطاری را نمایش دهد. بعد از تشخیص درست بودن دستور ورودی، وارد تابع مورد نظر میشویم.



**توابع**

int add\_matrix(string , struct store\*)

تابع add\_matrix برای اضافه کردن ماتریس جدید طراحی شده

البته چون دستور add دارای انواع مختلفی هست در این تابع سازوکار برای تشخیص نوع دستور هم وجود دارد.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Int tyid(string, struct store\*)

این تابع برای تشخیص نوع داده ایی میباشد.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Bool is\_diagonal(string, struct store\*)

این تابع برای تشخیص قطری بودن به کار می رود.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Bool is\_triangular(string, struct store\*)

برای بالا یا پایین مثلثی بودن استفاده میشود.

Bool is\_identity(string, struct store\*)

همانی بودن ماتریس را نشان میدهد.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Bool is\_semmetrick(string, struct store\*)

برابر بود اعضا در دو طرف قطر را برسی میکند.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Bool inverse(string, struct store\*)

جا به جا کردن اعضای دو طرف قطر و ذخیره کردن آن.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Bool change(string, struct store\*)

تغییر یک خانه از آرایه با گرفتن سطر و ستون از ورودی.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Bool show(string, struct store\*)

نمایش کامل یک آرایه.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Bool delete\_matrix(string, struct store\*)

حذف کردن یک ماتریس به طور کامل.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Bool exit(struct store\*)

حذف آرایه های دو بعدی.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

در این برنامه از git و cmake هم استفاده شده، و همچنین پروژ در github آپلود شده.

با تشکر.