

زمان شی است ۲

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

شی گرایی همچنان مسئله مهمی برای سپهر هست و جدیداً با سربارگذاری عملگرها آشنا شده است برای همین تصمیم گرفته که به مسئله های شی گرایی قبلی سر بزند و تابع های هرکدام را که توانست با استفاده از عملگرها پیاده سازی کند و حال به سراغ کلاس `time` رفته است.

در این سوال ما می خواهیم برخی از توابع کلاس `time` که در سوال **زمان شی است** نوشتیم را با سربار گذاری عملگرها درست کنیم.

- می خواهیم با استفاده از عملگر `<<` یک شی از کلاس `time` را از ورودی بگیرد (برای اینکار می توانید یا فرمت مشخص خاصی مشخص کنید یا پیام مناسب دهید).

```
friend istream & operator >> (istream &in, Time &t){
    in >> SOMETHING;
    return in;
}
```

- می خواهیم با استفاده از عملگر `>>` شی از کلاس `time` را چاپ کند و همچنین بنویسد که این ساعت در موقعی از شبانه روز است طبق تابعی که قبلاً پیاده سازی کردید.

```
friend ostream & operator << (ostream &out, Time &t){
    out << SOMETHING;
    return out;
}
```

- با استفاده از عملگرهای `>` یا `<` یا `>=` یا `<=` بتوان ۲ شی از این کلاس را مقایسه کرد.

```
friend bool operator<(const Time& t1, const Time& t2){  
    bool b;  
    //do comparison  
    return b;  
}
```

- می‌خواهیم عملیات cast کردن را با استفاده از overloading پیاده سازی کنیم، به شکلی که بتوانیم کلاس زمان را به رشته تبدیل کنیم.
- همچنین عملگر + برای جمع دو زمان که یک شیء زمان را بعنوان ورودی می‌گیرد.

هیپولی خسته تر از هفته پیش

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

بیا روراست باشیم. من خستممممم!!

یک روز که هیپولی داشت خستگی در می کرد به این فکر افتاد که اگر هر نفر بخواد اسم یک سری توابع برای ورودی گرفتن ، خروجی گرفتن ، جمع و بخواهد تابع بنویسد و اسمی که دوست دارد را برایش بگذارد اینطوری اگر ما بخوایم از اون کلاس استفاده کنیم، همش باید کد اون شخص نگاه کنیم و خب خیلی خسته کننده است برای همین رفت جستجو کرد و با سربارگذاری عملگر ها (operator overloading) آشنا شد و تصمیم گرفت که کد ماتریسی که هفته پیش زد با استفاده از این روش بهینه تر بکند.

لیست توابعی که باید عوض شوند:

- به جای تابع `matrix_cin` از عملگر `<<` استفاده کنید
- به جای تابع `matrix_cout` از عملگر `>>` استفاده کنید
- به جای تابع ضرب و تفریق و جمع یک ماتریس با ماتریس دیگر را از `*` ، `-` ، `+` استفاده کنید
- همچنین عملگر های `*` ، `+=` و `-=` برای اینکه نتیجه محاسبات در خود ماتریس ذخیره کند
- به جای تابع ضرب و تقسیم یک عدد در یک ماتریس از علامت `*` ، `/` استفاده کنید
- به جای تابع `is_equal` از عملگر `==` استفاده کنید
- به جای تابع `is_not_equal` عملگر `!=` استفاده کنید
- استفاده کردن از عملگر `()` به صورت (ستون، ردیف) برای نشان دادن داده موجود در آن خانه
- همچنین از عملگر `++` و `--` استفاده کنید، به این صورت که همه مقادیر ماتریس را به علاوه ۱ یا منهای ۱ بکند (عملگر ها در هر ۲ حالت **پسوند و پیشوند** تعریف شده باشند)

موارد دیگر:

- به جای استفاده کردن از عملگر `(x,y)` عملگر `[y][x]` را پیاده سازی کنید و روش های مختلف را در قالب یک فایل PDF ارسال کنید.

ریاضی با طعم شی گرایی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

هیپولی تصمیم گرفته که برای اینکه حسابی *operator overloading* بلد بشه یک کلاس چندجمله ای درست کنه ولی خب دوست داره که بهش کمک کنید برای همین مشخصات کلاسی که می خواد درست کنه برای شما می نویسد تا به هیپولی کمک کنید.

کلاس چندجمله ای در این کلاس ضرایب یک چندجمله ای را در یک وکتور نگه می داریم:

- ضرایب همگی صحیح است
- درجه همه ضرایب حداکثر ۲۰ می باشد متد های کلاس چندجمله ای:
- سازنده پیش فرضی که تمام ضرایب را در ابتدا صفر قرار می دهد
- سازنده کپی که چندجمله ای را در یک چند جمله ای دیگر کپی می کند
- عملگر تک عملوندی "-" که یک چند جمله ای را قرینه می کند
- عملگر های جمع و تفریق و ضرب برای ۲ چند جمله ای را تعریف کنید
- عملگر های == ، >= ، > ، <= ، < را برای ۲ چندجمله ای تعریف کنید
- عملگر << برای ورودی گرفتن یک چند جمله ای به صورت فرم استاندارد و نسبت داد آن به یک شی چندجمله ای
- عملگر >> برای چاپ کردن چندجمله ای به صورت فرم استاندارد.

فرم استاندارد به صورت زیر می باشد:

...

$$6x^7 - 2x^2 + x - 3$$

$$19x^{17} + 34x - 78$$

نکات:

- در هنگام چاپ باید بر اساس درجه به صورت نزولی چاپ شود
- در هنگام ورودی نیازی به رعایت ترتیب نیست
- توان صفر به صورت عدد ثابت نشان داده می شود
- ضریب صفر نشان داده نمی شود
- توان یک در x^1 نشان داده نمی شود
- اگر ضریب +۱ یا -۱ باشد، ضریب را نشان نمی دهد جز در شرایطی که توان صفر باشد

موارد امتیازی:

- استفاده از regex برای گرفتن ورودی. (برای آشنایی بیشتر این لینک و این لینک را مشاهده کنید.)
- پیاده سازی عملگر های / برای تقسیم و % برای باقی مانده ۲ چندجمله ای