

کلاسی به نام Fraction طراحی کنید که برای نگهداری اعداد کسری استفاده می‌شود. مثلاً در قطعه کد زیر f برابر $\frac{3}{5}$ و g برابر $\frac{1}{4}$ خواهد بود.

```
Fraction f(3,5);
Fraction g(1,4);
```

برای این کلاس اعضای زیر را طراحی کنید:

۱. دو سازنده، یکی با دو پارامتر صحیح که صورت و مخرج کسر را مشخص می‌کنند و یکی با یک پارامتر صحیح که مخرج را برابر یک فرض می‌کند. به خطاهای ممکن توجه کنید.

۲. عملگر قرینه، -، که قرینه‌ی کسر را بر می‌گرداند.

۳. عملگر جمع، به سه شکل مختلف: جمع دو عدد کسری با هم، جمع عدد صحیح با عدد کسری و جمع عدد کسری با عدد صحیح.

۴. عملگر افزایش (+=) که یک کسر را به اندازه‌ی یک کسر دیگر افزایش می‌دهد.

۵. عملگر تساوی (==)، به شکل درست آن و نه مقایسه‌ی مستقیم صورت‌ها و مخرج‌ها

۶. عملگر درج در خروجی (<<) که کسر را به صورت a/b می‌نویسد که a و b به ترتیب صورت و مخرج کسر هستند. لزومی ندارد کسر به شکل ساده‌شده نوشته شود. کافیت مقدار کسر نوشته‌شده درست باشد.

همچنین ثابتی از نوع Fraction با نام HALF تعریف کنید که مقدار $\frac{1}{2}$ را داشته‌باشد.

مثال زیر یک استفاده‌ی نمونه از این کلاس را نشان می‌دهد:

```
1 #include <iostream>
2 #include "fraction.h"
3
4 int main() {
5     Fraction f(3,4);
6     Fraction g(2);
7     Fraction h = f + g;
8
9     cout << h << endl;      // prints 11/4 (or any other value equal to 11/4)
10
11     (h += HALF) += -f; // h is now equal to 10/4
12
13     Fraction e(5, 2);
14     if (h == e)
15         cout << "Just as expected!";
16     else
17         cout << "Somethings wrong!";
18     return 0;
19 }
```

نحوه‌ی تحویل

کد شما باید شامل یک فایل هدر با نام fraction.h باشد که کلاس Fraction و عملگرهای ذکرشده را اعلان کند. در کنار این فایل می‌توانید هر تعداد فایل منبع یا فایل هدر، با هر نامی داشته‌باشید. علاوه بر فایل‌های برنامه یک Makefile نیز تحویل می‌دهید. این فایل باید به شکلی نوشته‌شود که با اجرای دستور make یک کتابخانه‌ی پویا (Dynamic Library) با نام fraction تولید شود (یعنی نام فایل باید libfraction.so باشد).

برای آزمودن برنامه‌ی شما، ما فایل fraction.h را در فایل‌های منبع خود #include می‌کنیم و پس از ترجمه‌ی فایل‌ها، با استفاده از کتابخانه‌ی پویایی که Makefile شما تولید می‌کند، نتیجه‌ی ترجمه‌ی خود را لینک و اجرا می‌کنیم.

تقسیم‌بندی کد شما در فایل‌های هدر و منبع باید به گونه‌ای باشد که استفاده‌ی درست ما از کلاس Fraction باعث ایجاد خطاهای هنگام ترجمه، لینک و/یا اجرا نشود.
در صورتی که شماره‌ی دانشجویی شما ۸۱۰۱۹۰۱۲۳ است، برنامه‌ی خود، شامل فایل‌های fraction.h و Makefile را به صورت یک فایل به نام A5-90123.zip در محل مربوطه در CECM آپلود کنید.

دقت کنید

- همچنان به نکاتی که در انتهای تمرینهای قبلی آمده توجه کنید!
- شما باید برنامه را به صورت خواسته‌شده پیاده‌سازی کنید. به بزرگی و کوچکی و حروف، و نوع ورودی و خروجی عملگرها توجه کنید.
- لازم است عملگرهای مذکور را حتی‌الامکان به صورت اعضای کلاس Fraction تعریف کنید.
- عملگرها باید به نحوی تعریف شوند که در شرایط استفاده‌ی صحیح درست عمل کنند.
- رعایت سبک برنامه‌نویسی درست و تمیز بودن برنامه شما در نمره تمرین تأثیر دارد. لازم است سبک نوشتن برنامه مطابق با قواعد کتاب باشد. اینجا را ببینید:

<http://www.stroustrup.com/Programming/PPP-style-rev3.pdf>