# بانک نرم‌افزار — تمرین شمارهٔ چهار

مدرس: رامتین خسروی — موعد تحویل: ۲۸ اسفند ۱۳۹۰

بانک نرم‌افزار، محلی است در دانشگاه که دانشجویان می‌توانند از آن نرم‌افزار قرض بگیرند! برای قرض گرفتن نرم‌افزار تا یک هفته هزینه‌ای دریافت نمی‌شود، اما بعد از یک هفته، به ازای هر روز تأخیر، جریمه‌ای از شخص دریافت می‌شود. هنگامی که یکی از اعضای بانک برای گرفتن یا بر گرداندن CD یا DVD به بانک مراجعه می‌کند، مسؤول بانک نام عضو، نام نرم‌افزار و تاریخ را در یک لیست وارد می‌کند.

مسؤولین بانک علاقه دارند که از روی این لیست اولاً جریمه‌ی همه‌ی اعضا را محاسبه کنند و ثانیاً فهرستی از نرم‌افزارهای امانت‌گرفته‌شده به دست بیاورند. شما باید برنامه‌ای بنویسید که این کار را انجام دهد.

برای این کار لازم است پنج کلاس به نام‌های Bank، Date، Person، Disc و Borrow تعریف کنید. برای این کلاس‌ها، متغیرها و توابع عضو مناسب تعریف کنید. (سعی کنید اصول برنامه‌نویسی شیءگرا که در کلاس درس آموخته‌اید را به خوبی رعایت کنید.)

کلاس Borrow برای نمایش یک عملیات قرض گرفتن کتاب استفاده می‌شود. برای این کلاس، تعریف توابع عضو زیر کافی است:

```
Borrow(string disc, Date date);
Date get_date();
bool is_late(Date delivery_date);
string get_disc_name();
```

کلاس Person هم برای نگه‌داری اطلاعات یک عضو استفاده می‌شود. برای این کلاس توابع عضو زیر را تعریف کنید:

```
Person(string name);
string get_name();
int get_late_days();
void borrow(Disc* disc, Date date);
void deliver(Disc* disc, Date date);
```

## ورودی

ورودی لیستی است که توسط مسؤول بانک نرم‌افزار نوشته می‌شود. در اولین خط، ابتدا، $n$، تعداد رخدادها (تحویل یا دریافت نرم‌افزار) و سپس، $t$، میزان جریمه به ازای هر روز دیرکرد در تحویل قرار می‌گیرند. سپس در $n$ خط بعدی، در هر خط یک رخداد توصیف می‌شود. در ابتدای توصیف هر رخداد سه عدد صحیح نشان‌دهنده‌ی روز، ماه و سال روی دادن رخداد قرار می‌گیرند. فرض کنید این تاریخ‌ها مطابق تقویم شمسی بوده و مقادیر آن‌ها صحیح هستند. همچنین فرض کنید ماه اسفند همیشه ۲۹ روزه است و سال کبیسه نداریم! پس از تاریخ، نام عضو و بعد از آن نام نرم‌افزار داده می‌شوند. فرض کنید هر دو نام فقط از حروف الفبا تشکیل شده‌اند. همچنین فرض کنید هیچ دو عضو و یا هیچ دو نرم‌افزاری نام یکسان ندارند (یعنی از هر نرم‌افزار فقط یک نسخه داریم).

دقت کنید که با در نظر گرفتن یک رخداد نمی‌توان تشخیص داد که آن رخداد تحویل نرم‌افزار است یا دریافت نرم‌افزار، اما با در نظر گرفتن اطلاعات خطوط قبلی، این کار امکان‌پذیر خواهد بود. می‌توانید فرض کنید که اطلاعات از لحاظ منطقی صحیح هستند، یعنی مثلاً قبل از اینکه یک نرم‌افزار باز گردانده شود، عضو دیگری آن را قرض نمی‌گیرد.

## خروجی

در خروجی ابتدا در یک خط عبارت "Fines:" را چاپ کنید و در خطوط بعدی به ازای هر یک اعضا مطابق خروجی نمونه، نام و میزان جریمه‌ی او را نشان دهید. پس از آن در یک خط دیگر عبارت "Borrowed Discs:" را نمایش داده و در خطوط پس از آن فهرستی از کتاب‌هایی که هنوز باز گردانده نشده‌اند را، مشابه خروجی نمونه، بنویسید. ترتیب نمایش اعضا و نرم‌افزارها اهمیتی ندارد.

## نمونه ورودی

```
8 500
1 1 82 mj office
9 1 82 mj office
13 2 88 mj ubuntu
20 2 88 ali gparted
25 2 88 mj ubuntu
20 12 90 hassan ubuntu
27 12 90 ali pes
1 1 91 ali pes
```

## نمونه خروجی

```
Fines:
mj: 3000
ali: 0
hassan: 0
Borrowed Discs:
ubuntu
gparted
```

## نحوه‌ی تحویل

تعریف هر کلاس باید به شکل صحیح در دو فایل h. و cpp. تقسیم شود. هیچ یک از فایل‌های شما نباید شامل تابع main یا هر تابع غیرعضور دیگری باشد. ما برای تصحیح تمرین شما فایلی مشابه فایل زیر را به همراه فایل‌های شما ترجمه و لینک می‌کنیم. بنابراین لازم است کد شما به شکلی باشد که فایل زیر بدون اشکال ترجمه، لینک و اجرا شود.

```cpp
#include <iostream>
#include <string>
#include "bank.h"
#include "date.h"
using namespace std;

int main() {
	int n, penalty;
	cin >> n >> penalty;

	Bank bank(penalty);
	for (int i = 0; i < n; i++) {
		int day, month, year;
		string disc_name, member_name;
		cin >> day >> month >> year >> member_name >> disc_name;

		if (!bank.has_member(member_name))
			bank.add_member(member_name);
		if (!bank.has_disc(disc_name))
			bank.add_disc(disc_name);

		Date date(day, month, year);
		bank.register_event(member_name, disc_name, date);

	}
	bank.print_fines();
	bank.print_borrowed_discs();

	return 0;
}
```

شکل ۱: شکل کلی تابع main که باید توسط برنامهٔ شما قابل ترجمه و اجرا باشد.

همان طور که گفته‌شد شما باید به ازای هر یک از پنج کلاسی که تعریف می‌کنید دو فایل با همان نام و با پسوندهای h. و cpp. داشته‌باشید. کلیه‌ی حروف نام فایل‌ها باید کوچک باشند. علاوه بر این فایل‌ها یک Makefile برای برنامه‌ی خود

تهیه کنید که پس از ترجمه کردن فایل‌های منبع، یک کتابخانه‌ی ایستا (Static Library) با نام bank که شامل همه‌ی فایل‌های منبع باشد، تولید کند. ما برای اجرای برنامه‌ی شما، یک فایل مشابه فایل بالا، با نام main.cpp را در کنار فایل‌های شما قرار داده و دستورات زیر را اجرا می‌کنیم:

```
mj@mjtp-jr:$ make
mj@mjtp-jr:$ g++ main.cpp -l bank -L.
mj@mjtp-jr:$ ./a.out
```

پیش از تحویل مطمئن شوید که برنامه‌ی شما کلیه‌ی مراحل بالا را بدون اشکال پشت سر می‌گذارد.
در صورتی که شماره‌ی دانشجویی شما ۸۱۰۱۹۰۱۲۳ است، ۱۱ فایل شامل ۱۰ فایل هدر و منبع و یک Makefile را به صورت یک فایل به نام A4-90123.zip در محل مربوطه در CECM آپلود کنید.

## دقت کنید

- به نکاتی که در انتهای تمرینهای قبلی آمده توجه کنید!
- شما باید برنامه را به صورت خواسته‌شده پیاده‌سازی کنید. یعنی کلاس‌های مذکور را تعریف کرده و با ایجاد کردن objectهایی از آنها و فراخوانی توابع عضو آنها به صورتی که گفته‌شد به حل مسأله بپردازید.
- رعایت سبک برنامه‌نویسی درست و تمیز بودن برنامهٔ شما در نمرهٔ 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