دانشگاه صنعتی اصفهان

نیمسال دوم سال تحصیلی۴۰۱ -۴۰۰

سربرگ تمرین: IUTAP4002E05

فرض کنید برای انسان‌ها و دانشجویان دو ساختمان زیر تعریف شده باشند.

```
struct Human
{
        char name[30];
        char * family;
        int n_code; //شماره ملی
};
```

```
struct Student
{
        Human * h;
        int std_no; // شماره دانشجویی
        char field[50]; // رشته تحصیلی
};
```

۱. این موجودیت‌ها را به صورت کلاس در بیاورید. تمام اقلام داده‌ای باید private باشند و هر جا لازم است از set و get متناسب استفاده شود. سازنده و مخرب را برای این کلاس‌ها پیاده‌سازی کنید تا متوجه شوید هر جا منبعی پویا باید اخذ شود مدیریت اخذ و رهاسازی آن با شماست. برای این تمرین این کارها را الزاماً در سازنده مخرب انجام دهید. برای کلاس Student یک سازنده ۵(؟!) پارامتری هم بسازید که مقادیر مورد نیاز آن را در همان زمان ساختن شیء در اختیار کلاس قراردهد. اما برای Human این کار را صرفاً از طریق توابع set انجام دهید تا ببینید که یک سازنده چند پارامتری میتواند در فراخوانی چند تابع صرفه جویی کند. سازنده ساده برای کلاس Human باید نام را به "ensane" و فامیل را به "pishfarz" و کدملی را به ۱۲۳۴۵۶۷۸۹۰ مقداردهی کند.

۲. در کلاس، کدهای مربوط به لیست پیوندی را برایتان گفتیم. در آن مثال، لیست پیوندی را یک‌طرفه ساختیم. کلاس‌ها و تمام توابع مربوطه را طوری پیاده‌سازی کنید که لیست پیوندی دو طرفه شود و نوع داده‌ی آن از نوع Student باشد. دو طرفه بودن لیست به معنی آن است که هر عضو لیست، عضو قبلی را نیز بداند.

۳. تابع print را برای هر ۴ نوع CLinkedList، CNode، Student و Human تعریف کنید. در این صورت تابع print که در LinkedList بود باید مکرراً تابع print برای CNode را فراخوانی کرده و این تابع نیز تابع print مربوط به Student را فراخوانی کند. این تابع علاوه بر فراخوانی تابع print مرتبط با Human سایر اطلاعات دانشجو را نمایش می‌دهد. print برای Human تمام اطلاعات آن انسان را نمایش می‌دهد.

۴. تابع pop را اضافه و آن را طوری پیاده کنید که داده گره اول لیست را برگرداند. سپس آن گره را از لیست حذف کند.

۵. برنامه شما باید ساختار چند فایلی داشته باشد و تمام کلاس ها باید سازنده و مخرب داشته باشند.

۶. در main یک لیست پیوندی دو طرفه بسازید و تعدادی دانشجو را به آن اضافه و چاپ کنید. سعی کنید متوجه شوید کجا سازنده و مخرب برای هر کلاس فراخوانی می‌شود.

۷. لازم نیست به این سوال جواب دهید اما به این فکر کنید چرا Student has a Human؟ آیا Student is a Human درست‌تر نیست؟ برای پاسخ به این نیاز چه مفهومی در برنامه نویسی وجود دارد؟

پیروز باشید.