



تکالیف خود را تا تاریخ معین شده به آدرس [mostafa.mahdjou@gmail.com](mailto:mostafa.mahdjou@gmail.com) ارسال نمایید. دقت کنید که عنوان ایمیل خود را با این الگو منطبق کنید: `tmu_ap_[exercise number]`؛ بطور مثال برای تمرین اول می‌بایست این عنوان را استفاده نمایید: `tmu_ap_1`.

همچنین برای ارسال تکالیف، فقط و فقط فایل `.cpp` مرتبط با هر تمرین را و نه فایل پروژه‌ی ساخته شده در Visual Studio را، در یک پوشه قرار داده و آن را بصورت RAR شده ارسال نمایید. نکته‌ی مهم دیگر در ارتباط با نام فایل‌های `.cpp` ارسال شده است. فایل‌های `.cpp` ارسال شده می‌بایست با الگوی `studentNumber_[exerciseNumber]_[arbitraryFileName].cpp` نامگذاری شوند. همچنین منظور از `arbitraryFileName` نام دلخواهی است که برای فایل مرتبط با آن تمرین می‌خواهید در نظر بگیرید. برای مثال برای نامگذاری فایل مرتبط با پاسخ سوال اول، تکالیف سری نخست، با فرض اینکه `studentNumber` اختصاصی فرد ارسال کننده ۵ باشد، بدین صورت خواهد شد: `_5_1_somefilename.cpp`.

لازم به ذکر است، محدودیتهای گذاشته شده در خصوص عناوین ایمیل‌ها و همچنین نام فایل‌ها به جهت انجام مرحله‌ی تست نرم افزاری جهت بررسی کدهای کپی برداری شده می‌باشد. همچنین در هر مرتبه، تمرین تعدادی از دانشجویان بصورت تصادفی، علاوه بر ارسال بصورت ایمیل، به شکل حضوری تحویل گرفته خواهد شد. در صورت ناتوانی در پاسخگویی و یا تشخیص کپی برداری به هر نحو توسط تحویل گیرندگان تمرین، عواقب خوبی در انتظار دانشجوی مربوطه نخواهد بود.

**توجه :** ارسال تمرین با تأخیر مشمول کسر نمره خواهد شد.



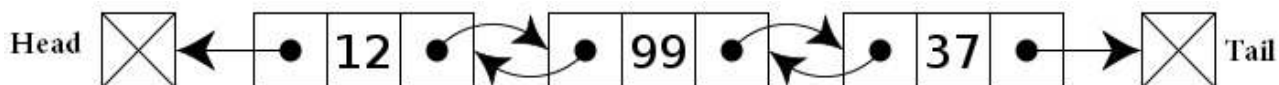
## ۱. مرتب سازی ادغامی<sup>۱</sup>:

برنامه‌ای بنویسید که به کاربر این امکان را بدهد که تعدادی عدد، که تعداد دقیق آنها را کاربر، پیش از وارد نمودن اعداد، مشخص خواهد نمود، وارد کرده، و سپس اعداد وارد شده را بصورت بازگشتی و بر اساس الگوریتم مرتب سازی ادغامی، مرتب کرده و نتیجه را در خروجی چاپ نماید.

\*راهنمایی: برای گرفتن ورودی از تخصیص حافظه بصورت پویا بهره بگیرید.

## ۲. لیست پیوندی دو طرفه<sup>۲</sup>:

لیست پیوندی دو طرفه شبیه لیست پیوندی یک طرفه است، با این تفاوت که هر عنصر علاوه بر داشتن اشاره گر به عنصر بعدی خود، اشاره گری به عنصر قبلی خود نیز دارد. ساختار آن به شکل تصویری بصورت زیر است:



برنامه‌ای بنویسید که مشخصات تعدادی دانشجو را که پایان آنها با NULL مشخص می‌گردد را از ورودی گرفته و در هر لحظه که مشخصات یک دانشجو وارد لیست می‌شود، لیست بصورت مرتب شده بر اساس شماره‌ی دانشجویی باقی بماند. یعنی اطلاعات هر دانشجو به هنگام ورود، می‌بایست در جای مناسب خودش در لیست قرار بگیرد. به این شیوه‌ی دریافت ورودی و مرتب سازی، ورود مرتب شده<sup>۳</sup> می‌گویند. این برنامه همچنین، می‌بایست امکان نمایش اطلاعات دانشجویان حاضر در لیست، حذف اطلاعات یک دانشجوی خاص با دادن شماره‌ی دانشجویی، و امکان افزودن اطلاعات جدید به لیست را داشته باشد. بهتر است برای برنامه‌ی خود، منویی شبیه به تصویر زیر فراهم کنید:

1. Enter students
2. Delete a student
3. Print students
4. Exit

اطلاعات هر دانشجو شامل نام و نام خانوادگی، شماره‌ی دانشجویی و رشته‌ی تحصیلی است.

<sup>۱</sup> Merge Sort

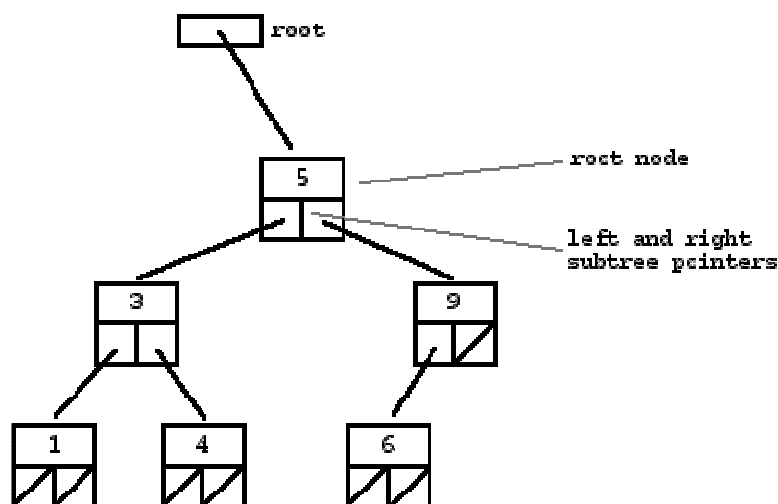
<sup>۲</sup> Doubly Linked List

<sup>۳</sup> Sorted Insert



### ۳. تبدیل درخت دودویی مرتب<sup>۴</sup> به لیست پیوندی (تشویقی)

درخت دودویی ساختار داده‌ای است که هر عنصر آن دارای یک بخش داده، و دو بخش دیگر برای اشاره به فرزندان چپ و راست آن است که آنها نیز به نوبه‌ی خود یک درخت دیگر محسوب می‌شوند. در واقع هر گره فقط و فقط دارای دو فرزند است. شکل این ساختار را در تصویر زیر می‌توانید مشاهده نمایید.



در واقع این ساختار شباهتهایی به لیست پیوندی دارد، از این جهت که هر عنصر حاوی اشاره‌گری به عناصر مرتبط با آن است.

درخت دودویی مرتب، درخت دودویی‌ای است که کلیه‌ی عناصر سمت چپ هر گره کوچکتر از آن و کلیه‌ی عناصر سمت راست هر گره بزرگتر از آن می‌باشند.

در این سوال شما می‌بایست علاوه بر پیاده سازی ساختار داده‌ی درخت، و وارد نمودن تعدادی عنصر در آن، برنامه‌ی بازگشتی‌ای بنویسید که ساختار داده‌ی درخت را بصورت یک لیست پیوندی دو طرفه در آورده و اطلاعات عناصر آن را در خروجی چاپ نماید.

\*درست است که این سوال بصورت تشویقی در نظر گرفته شده، اما به دلیل ترکیبی بودن این سوال که شامل مباحث بازگشتی، اشاره گر و همچنین لیست پیوندی، می‌باشد، موجب فهم عمیق شما از این مباحث خواهد شد. پس حل این سوال اکیدا توصیه می‌شود.