

بسمه تعالی

تمرین برنامه نویسی ۲

مهلت برای تمامی گروه ها: جزئیات تاریخ تحویل در گروه اعلام شده است

- در این تمرین، به هیچ عنوان از **آرایه** استفاده نکنید. ساختمان داده ای غیر از آرایه انتخاب و پیاده سازی کنید، و دلیل انتخاب خود را در گزارشی که ضمیمه برنامه خواهید کرد، شرح دهید.
- زبان برنامه نویسی می تواند یکی از زبان های C، C++ و یا جاوا باشد.
- در صورت نوشتن برنامه با زبان C بایستی از Struct ها و Pointer ها استفاده شود.
- در صورت نوشتن برنامه با زبان C++ بایستی از Template استفاده شود.
- در صورت نوشتن برنامه با زبان Java بایستی از Generic Classes استفاده شود.
- برنامه نویسی به زبان C++ نمره مثبت خواهد داشت.
- این تمرین حاوی یک گزارش نیز خواهد بود که در آن موارد خواسته شده در سوالات پاسخ داده شده است.
- برنامه خود را به همراه گزارش به آدرس ایمیل aut_ds@yahoo.com با عنوان زیر ارسال نمایید.

[StudentNO]

تمرین: فرض کنید در کلاس درس ساختمان داده، استاد درس قصد دارد برنامه ای برای مدیریت نمرات دانشجویان بنویسد. به سوالات زیر پاسخ دهید:

(نکته: هر رکورد شامل مجموعه ای از اطلاعات مرتبط با یک دانشجو (از قبیل نام، شماره دانشجویی، جنسیت و نمره دانشجو در درس ساختمان داده) است. تمام رکوردها دارای نمره هستند، و چندین رکورد ممکن است دارای نمره یکسانی باشند.)

الف) برنامه ای بنویسید که استاد درس بتواند نمرات دانشجویان را در آن وارد و نگهداری کند. این برنامه اطلاعات دانشجویان را به همراه نمره وی دریافت کرده و به صورت مرتب شده (بر اساس نمره) ذخیره می کند.

ب) فرض کنید استاد درس بخواهد اطلاعات دانشجویانی را که نمره آنها X است، مشاهده کند. علاوه بر این قصد دارد اطلاعات دانشجویانی که نمره آنها نزدیک به نمره X است را نیز مشاهده کند. به عبارت دیگر، برنامه شما نمره ای را دریافت

می کند و لیست اطلاعات دانشجویانی که نمره آنها برابر با نمره دریافتی است، نمایش می دهد، سپس منتظر وارد کردن یکی از دستورات زیر از جانب استاد درس خواهد بود:

- Next: بعد از وارد کردن کلمه Next، برنامه شما بایستی اطلاعات دانشجویانی را که نمره آنها بلافاصله بعد از x است، نمایش دهد.
- Prev: بعد از وارد کردن کلمه Prev، برنامه شما بایستی اطلاعات دانشجویانی را که نمره آنها بلافاصله قبل از x است، نمایش دهد.
- Enough: بعد از وارد کردن کلمه Enough، اجرای برنامه خاتمه می یابد.

به عنوان مثال ورودی های زیر را در نظر بگیرید:

{ (A,11), (B,11), (C,12), (D,14), (E,14), (F,15), (G,17), (H,17), (I,17), (J,18) }

با وارد کردن عدد ۱۴، مقادیر زیر در خروجی نمایش داده خواهد شد:

D, E

با وارد کردن کلمه Next مقادیر زیر در خروجی نمایش داده خواهد شد:

F

با وارد کردن دوباره کلمه Next مقادیر زیر در خروجی نمایش داده خواهد شد:

G, H, I

با وارد کردن کلمه Prev مقادیر زیر در خروجی نمایش داده خواهد شد:

F

و با وارد کردن کلمه Enough اجرای برنامه به اتمام می رسد.

برنامه شما بایستی Next و Prev را با $O(1)$ انجام دهد. ساختمان داده طراحی شده در بخش الف را تغییر دهید تا شامل موارد خواسته شده بخش ب نیز باشد. تغییرات را به همراه دلایل ذکر کنید.

ج) برای افزایش عملکرد برنامه خود، بخش دیگری به آن اضافه کنید. اگر تعداد دانشجویان بسیار زیاد باشد، جستجو از ابتدای لیست می تواند زمانگیر باشد. بنابراین پس از آنکه برنامه شما جستجویی را انجام داد، منتظر دریافت دستور جستجوی بعدی نیز باشد، و جستجوی جدید را از نقطه ای از لیست انجام دهد که در حال حاضر در آن قرار دارد. فرض کنید جستجو برای نمرات عمدتاً به صورت یکی از حالات صعودی یا نزولی انجام می گیرد. برنامه خود را طوری تغییر دهید که مطابق با دو مد صعودی یا نزولی جستجوی خود را انجام دهد. به عبارت دیگر در مثال فوق اگر جستجو در مد

نزولی باشد، اگر بعد از جستجوی ۱۸ و مشاهده خروجی، نمره ۱۲ وارد شود، جستجو (به جای شروع از ابتدای لیست) از مکان ۱۸ به صورت نزولی خواهد بود.

(لیست پیوندی حلقوی می تواند برای چنین رویکردی مناسب باشد. تا در صورت رسیدن به ابتدای لیست، جستجو را به راحتی از انتها آغاز کند.)

در این بخش از سوال، چرا استفاده از لیست حلقوی به جای جستجو از ابتدا می تواند باعث افزایش سرعت برنامه شود؟

موفق باشید

ابراهیم پور