

گزارش تمرین سری 6 ام برنامه نویسی پیشرفته

علیرضا طباطبائی

سوال 1 :

در این سوال ، ورودی ها را بصورت **template** تعریف کردیم که همه ی 4 تست را بتوان توسط یک تابع پاس کرد.

همچنین چون در تست 4 ، ورودی سوم نداشتیم ، پس باید مقداری **default** را برای آن معین میکردیم.

همچنین بایستی در حلقه **while** ، شرطی برای توقف الگوریتم تعیین میکردیم که اگر میزان تغییر در هر مرحله نسبت به مرحله قبل از عددی خاص کمتر شود ، الگوریتم متوقف شود زیرا همگرا شده است.

در نهایت مشتق را نیز بصورت اختلاف  $y$  به  $x$  تعریف کردیم که دو نقطه نزدیک به هم را برای این کار انتخاب نمودیم.

## سوال 2 :

ابتدا فایل را خوانده و سپس **regular expression** مورد نظر را تعیین کردیم .  
سپس در فایل ، یک جستجو انجام دادیم و تمام نمونه ها را وارد بردار کردیم و سپس با استفاده از **lambda function** و تابع **sort** ، آنها را مرتب کردیم.  
در لاندای ورودی ها را رفرنس بگیریم زیرا کلاس **patient** دارای **copy constructor** نیست.  
ورودی های تابع **std::sort** نیز باید بصورت **iterator** باشند.

## سوال 3 :

این سوال نیز همانند سوال دوم میباشد.  
با این تفاوت که داده ها در **priority\_queue** ذخیره میشوند و باید از قبل تابعی که برای مرتب سازی آنها میباشد را تعریف نماییم.  
این کار را نیز بواسطه **lambda** انجام دادیم.

## سوال 4 :

در این سوال ابتدا باید 2 کلاس را همراه با `constructor` های آنها تعریف نماییم و سپس سراغ تعریف فیلتر `Kalman` میرویم : برای حلقه زدن بر روی کل داده ها در فیلتر باید از `for_each` استفاده نماییم.

ورودی ها بصورت `iterator` میباشد و تابعی که بر روی هر داده اعمال میشود نیز ورودی سوم است که بصورت `lambda` میباشد.

نکته قابل توجه آنست که در تابع `lambda` باید یک استراکچر که برای تعریف نقطه میانگین وزندار است را `capture` نماییم و این کار باید بصورت رفرنسی انجام شود. همچنین چون در نهایت باید بر مجموع کل وزن ها تقسیم نماییم نیز میبایست این مجموع وزن ها را کپچر نماییم.

# لینک گیت هاب