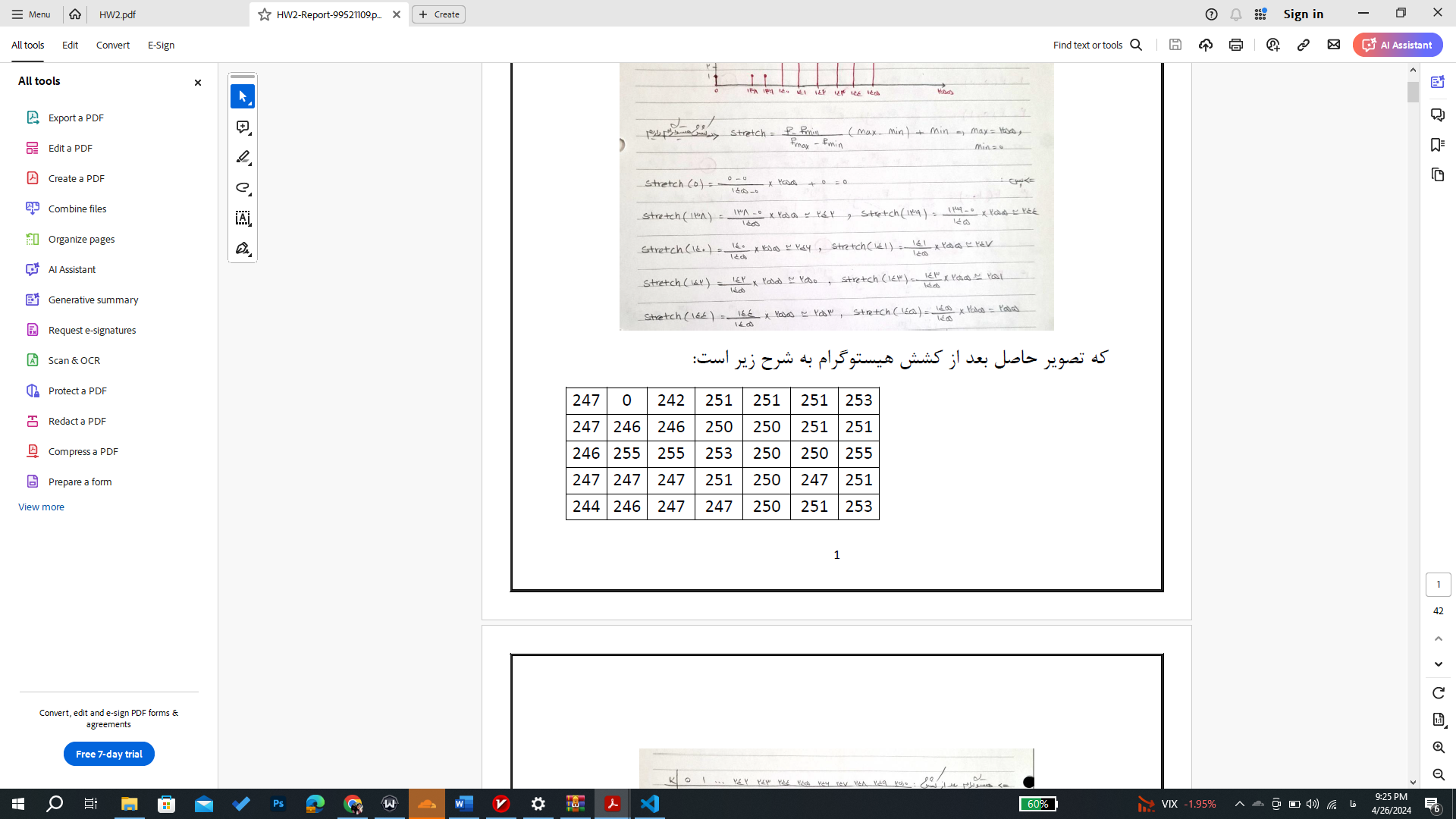
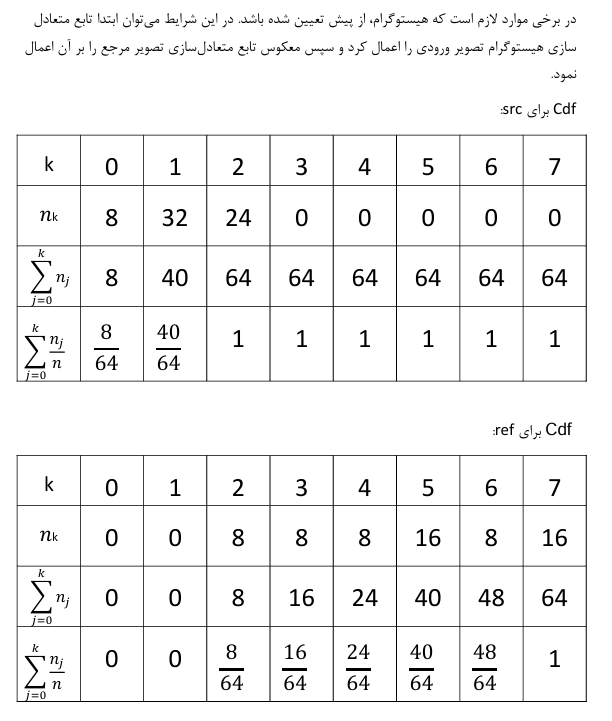
1. الف)

تصویر بعد از کشش هیستوگرام به شرح زیر است:





1. الف)



1. الف) چون در بیشتر نقاط تصویر روشنایی کم می باشد با استفاده از histogram equalization روشنایی به طور کلی زیاد می شود. این اتفاق برای نقاط تاریک صفحه مناسب است اما همه نقاط صفحه لزوما تاریک نمی باشند مثلا سر مجسمه که در ابتدا نیز روشن است، روشنایی بیشتری پیدا میکند و وضوح و شفافیت در بخشی مثل سر مجسمه به مقدار زیادی کاهش پیدا میکند، پس این روش نمی تواند به طور کلی روش مناسبی باشد.

ب) با توجه به صورت سوال برای ace 2 روش وجود دارد روش اول آن که عکس را به صورت پنجره به پنجره در گرفته و برای هر پنجره histogram equalization را پیاده سازی میکنیم و روش دوم آن است که برای هر نقطه با توجه به همسایه ها مقداری padding در بالا ، راست و پایین و چپ در نظر گرفته شود و با توجه پیکسل های همسایه تابع تبدیل نوشته شود که در روش اول با توجه به آن که هر پنجره ممکن است پیکسل های متفاوتی داشته باشد، این تفاوت می تواند باعث به وجود آمدن نقاط مرزی میان عکس ها شود که تصویر از یک دست بودن خارج شود(مثلا برای نقاط مرزی نزدیک به سر مجسمه این اتفاق کاملا مشهود است) و در این صورت تصویر یک دست بودن خود را از دست می دهد و پیوستگی ندارد در روش دوم نیز در بخش هایی که همه نقاط مقادیری نزدیک به یک دیگر داشته باشند (مثلا گوشه بالا سمت چپ تصویر) و این مورد باعث می شود که معادله تبدیل شارپی داشته باشیم که مثلا پیکسل های از 0 تا 10 را به 0 تا 255 مپ کند و همین باعث بروز مشکل شود. در روش clahe اما تعداد زیاد پیکسل ها مدیریت می شود و اگر مقدار پیکسل های یک بخش به مقدار خاصی زیاد باشند با استفاده از clip limit در نظر گرفته شده هیستوگرام را از مقدار خاصی به بعد برش میزند و مقدار های برش خورده را به صورت یکنواخت میان همه پیکسل ها پخش می کند و با این کار نویز هایی که در بخش قبل به آن اشاره شد به خوبی حذف می شوند که در تصویری که کد ها قرار داده شده این قضیه به طور کامل مشهود است، روند کاری ما نیز با توجه به شکل پایین صورت می گیرد:

ج) در این بخش کد مقادیر مختلفی برای grid و clip limit در نظر گرفته شده است، هنگاهی که پنجره بزرگ می شود، میخواهیم که تابع تبدیل را با توجه به همسایگی بزرگی مشخص کنیم و درتصویر نقاطی وجود دارد که خیلی روشن و تاریک باشند و همین باعث می شود که خروجی دقیقی را دریافت نکنیم اما در این حالت نیز که پنجره بزرگی داریم مثلا 128 که در سوال آمده، اگر clip limit مقدار کمتری باشد باعث می شود که تعداد پیکسل هایی که از یک مقدار بیشتر می شود پخش شود. پس در این حالت نیز وقتی clip limit، 2 هست خروجی بهتری نسبت به زمانی که 128 هست دریافت می کنیم، در صورتی که پنجره در نظر گرفته شده کوچک تر باشد، مثلا 16 که در سوال آمده با توجه به توضیحات بالا تصویر دقیق تر از حالت 128 تایی شده و اگر clip limit با مقدار کمتر را در نظر بگیریم متوجه می شویم که در 4 حالت مورد بررسی حالتی که clip limt = 2 , gridsize = 16 باشد بهترین تصویر را خواهیم داشت.

1. .
2. .
3. 