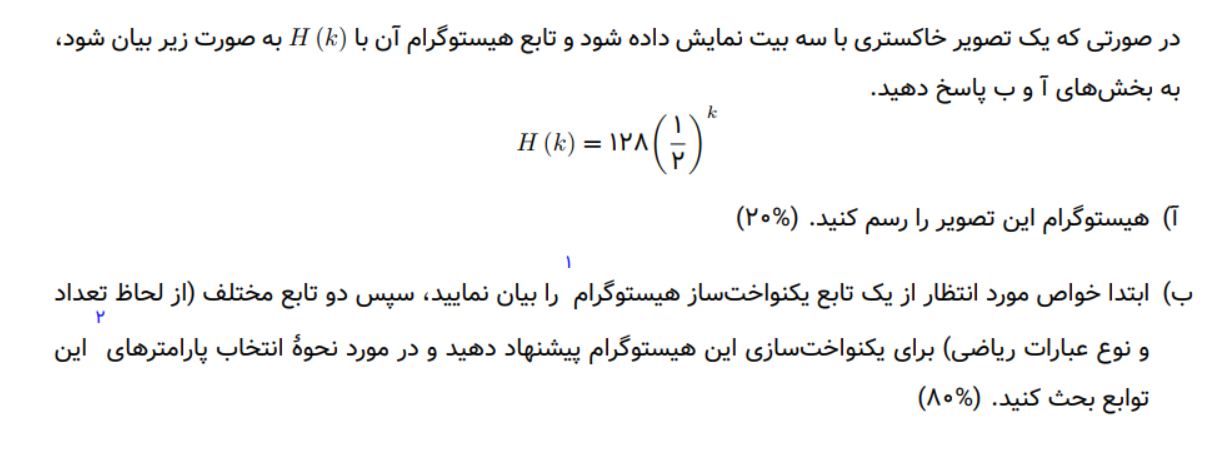
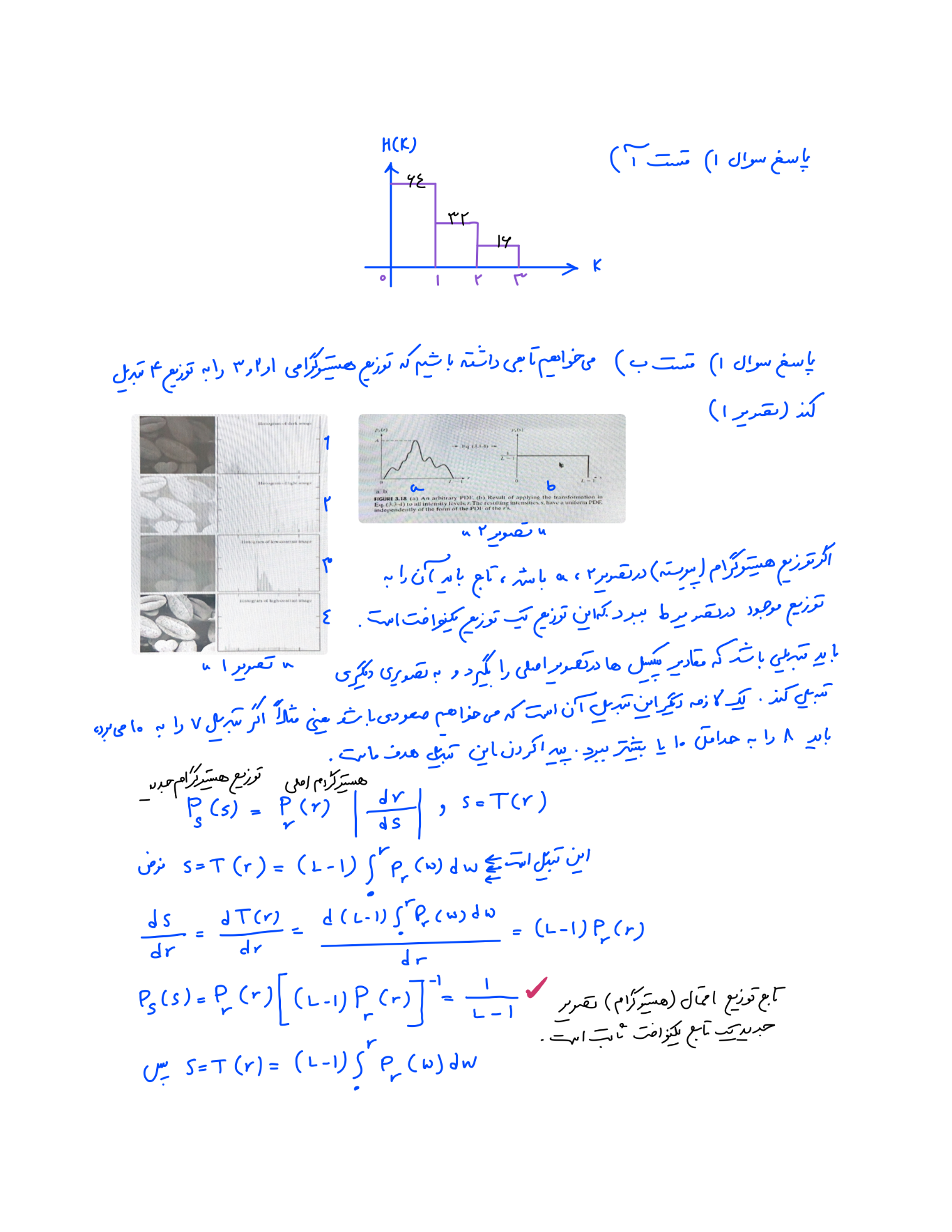
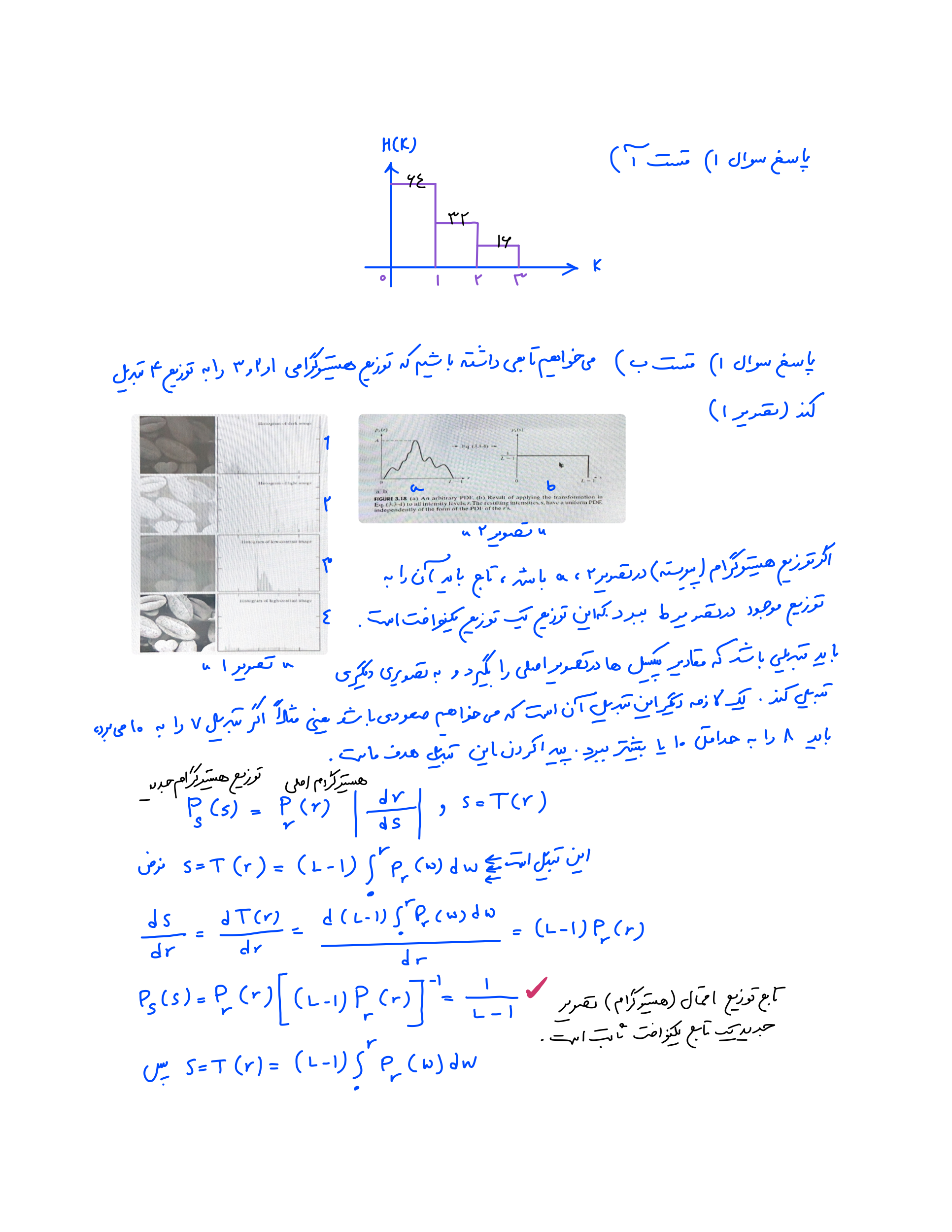
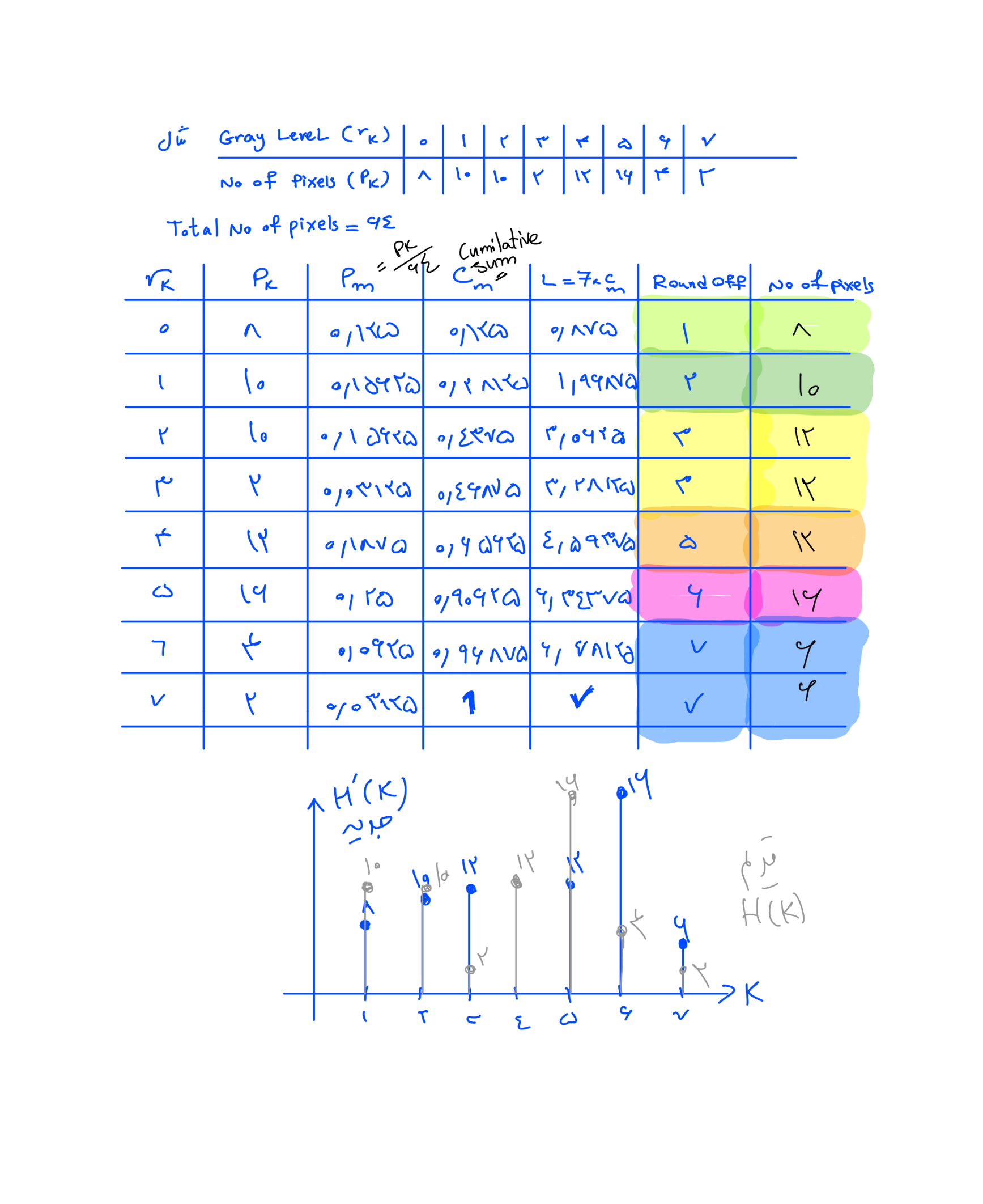
# سوال 1



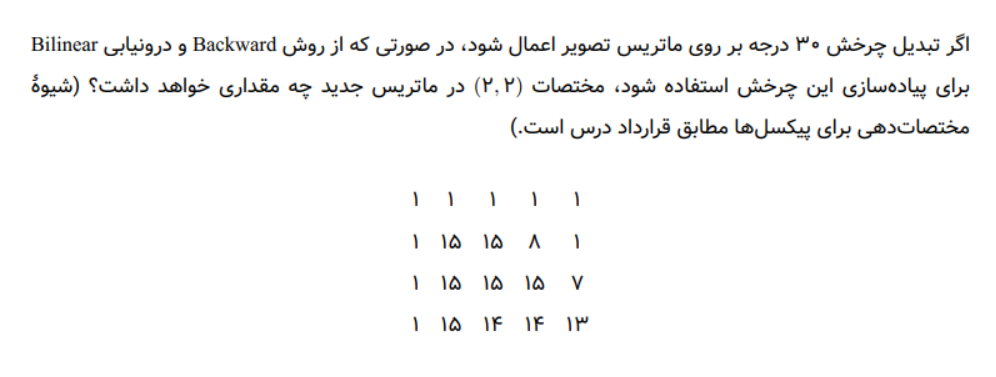


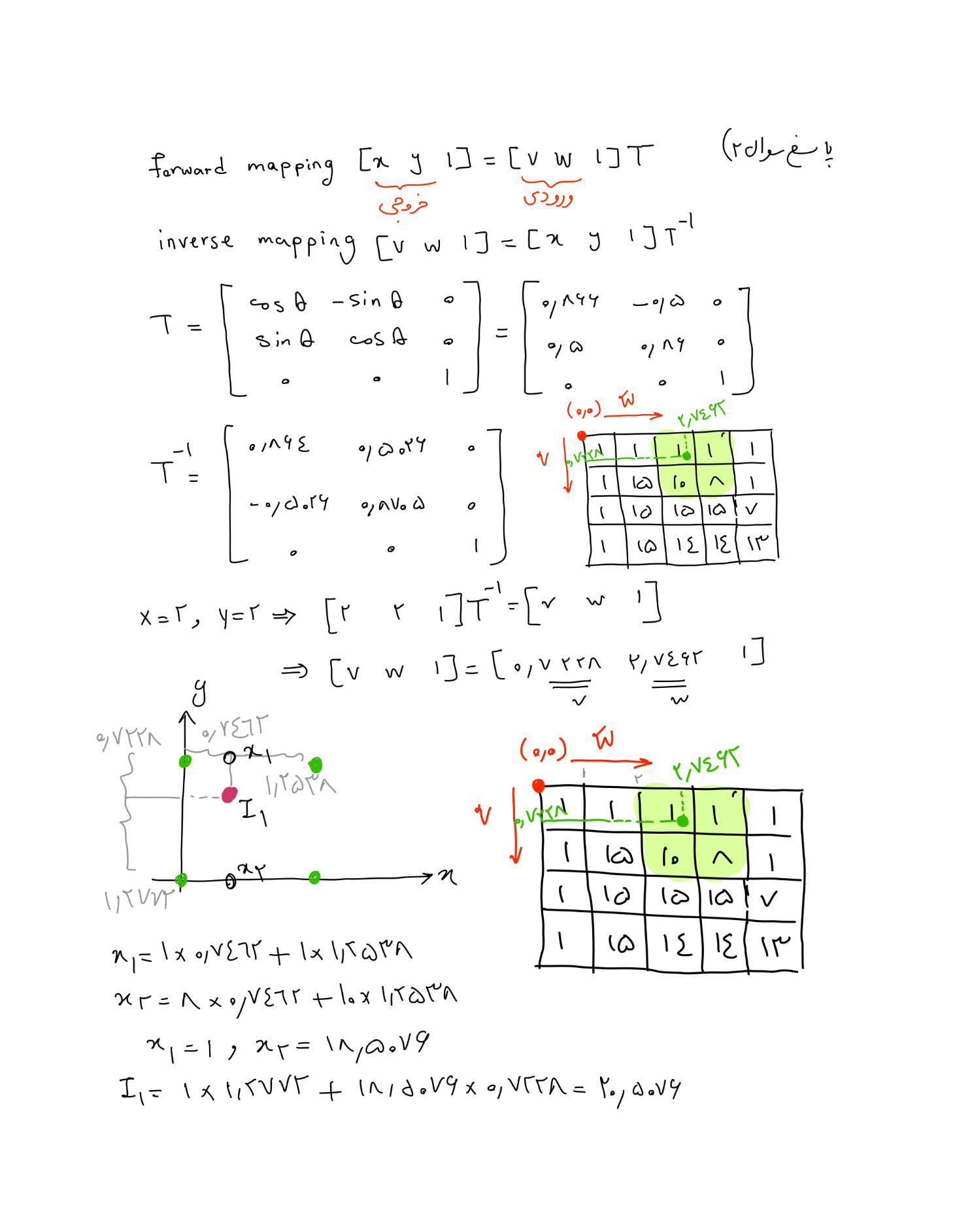


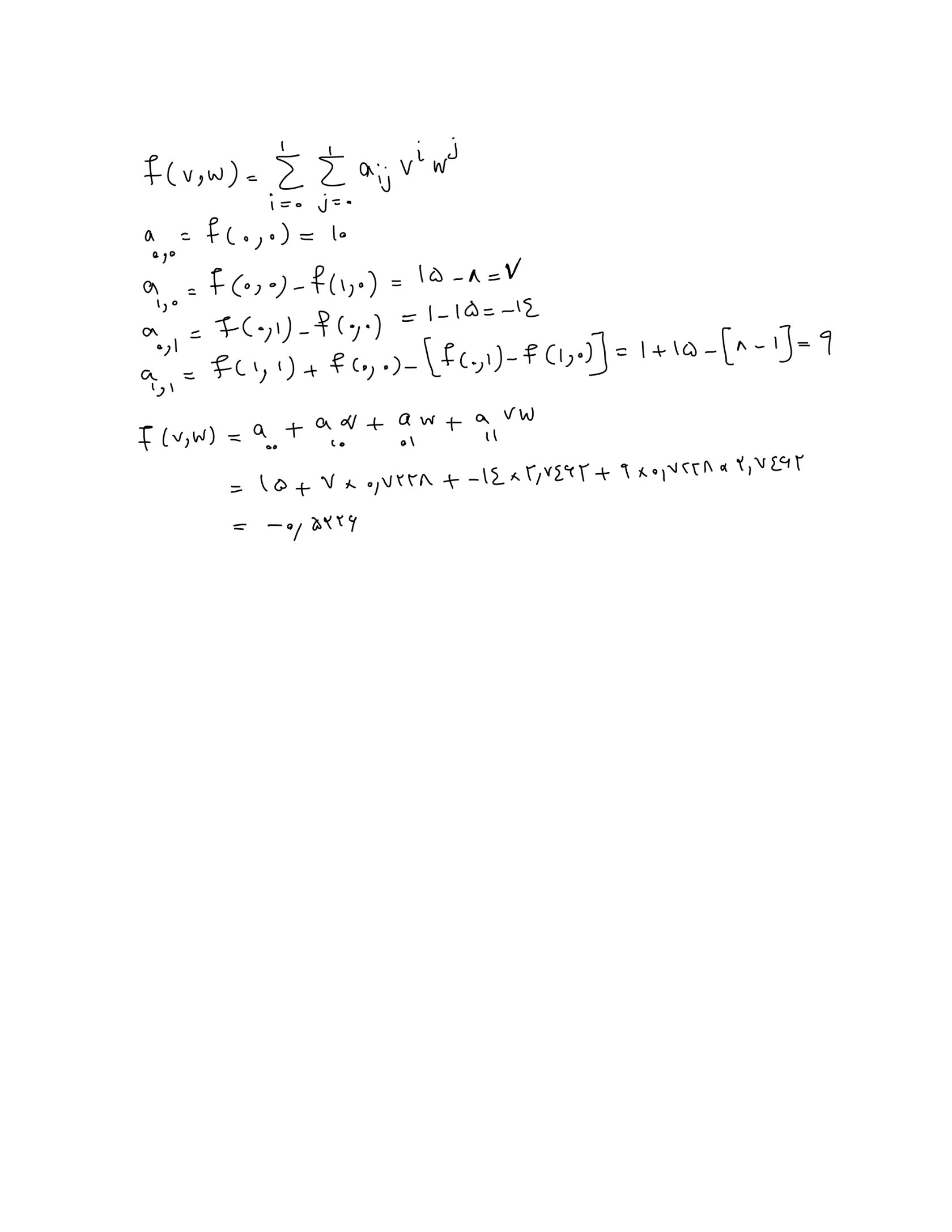


# 

# سوال 2







# سوال 3

## پاسخ الف)

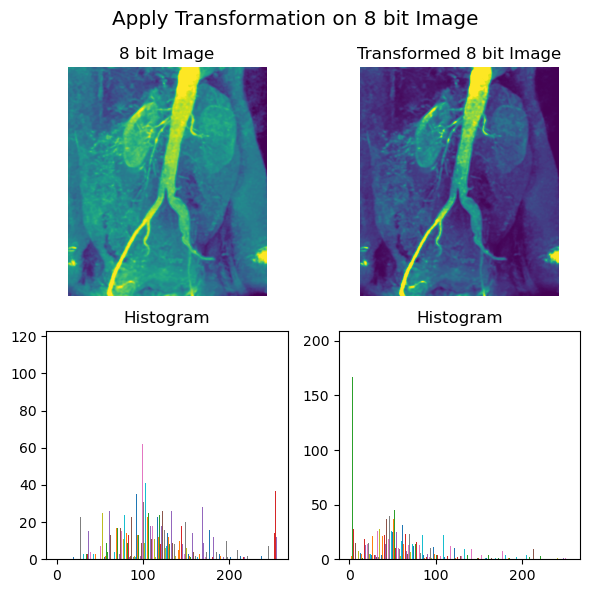
a را برابر معکوس 2 به توان تعداد بیت های تصویر در نظر گرفتم. به این ترتیب هیچ‌گاه شدت هر پیکسل پس از اعمال تبدیل بر روی آن پیکسل بیش از حد مجاز (2 به توان تعداد بیت تصویر) نمی‌شود.

## پاسخ ب)

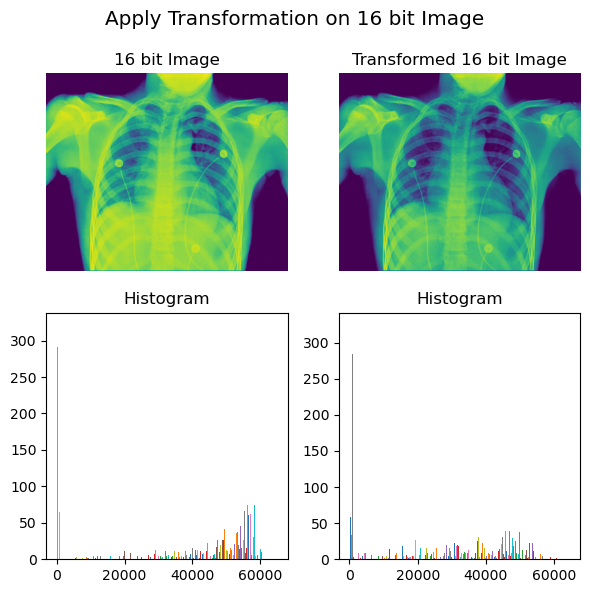
تابع خواسته شده را نوشتم. این تابع روی یک تصویر با تعداد بیت های مشخص به شکل دیجیتال اعمال می‌شود. به این ترتیب که ابتدا نوع داده پیکسل ها را به float32 تغییر دادم تا ظرفیت تبدیل شدن و بزرگ تر شدن را داشته باشند. سپس تابع تبدیل را روی آنها اعمال کردم و 0بعد از آن تقسیم بر 2 به توان تعداد بیت ها نمودم. سپس گردشان کردم. در آخر هم با توجه به این که بزرگترین مقداد همچنان مانند تصویر اولیه است، نوع داده پیکسل های خروجی را به نوع داده ورودی برگرداندم.

## پاسخ ج)

تابع را روی تصویر 8 بیتی اعمال کردم. خروجی آن به شکل زیر شد:

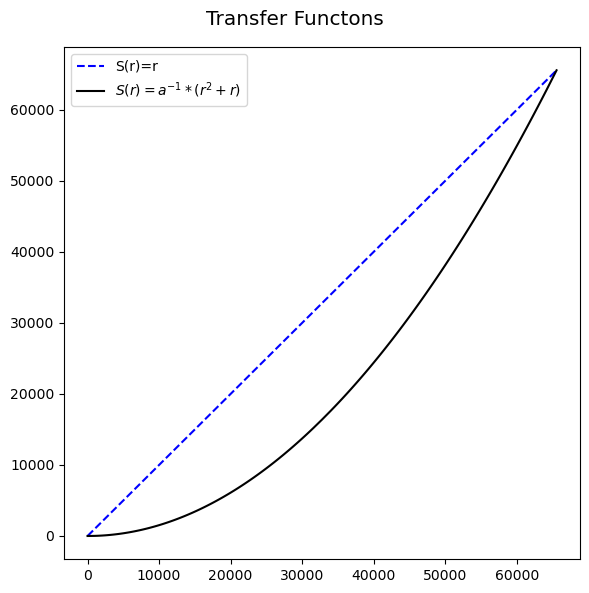


تابع را روی تصویر 16 بیتی هم اعمال کردم. خروجی آن به شکل زیر شد:



## پاسخ د)

در این قسمت تابع همانی به شکل خط چین آبی و تابع تغریف شده در بالا را به شکل خط پیوسته مشکی نمایش دادم. فرض کردم که روی یک تصویر 16 بیتی آن را اعمال می‌کنم و به این معنی که محور x ها از صفر تا 2 به توان 16 منهای یک بصورت گسسته تقسیم بندی شده است.



## پاسخ ه)

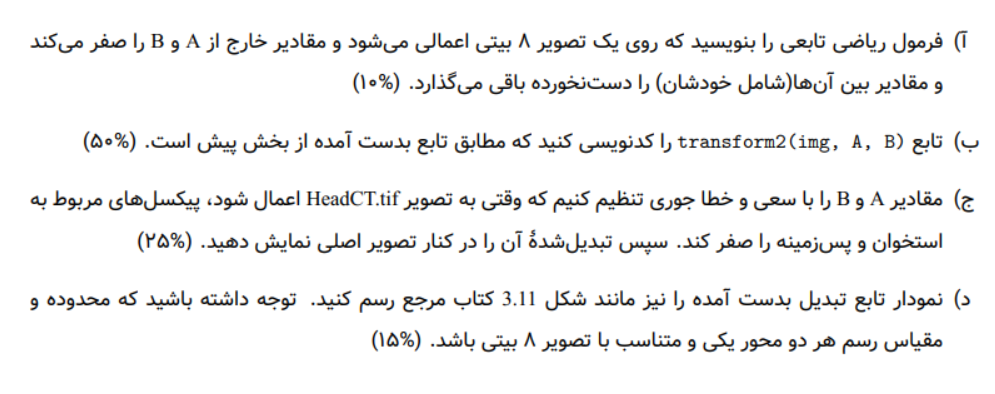
(ا) این تابع تبدیل برای بهبود تصاویر روشن مناسب است. زیرا مقادیر روشن را تاریک تر می‌کند.

(ب) تبدیل لگاریتمی بصورت زیر:



این تبدیل مقادیر خیلی تاریک را روشن تر ‌می‌کند. پس برای تصاویر تاریک مناسب است.

# سوال 4)

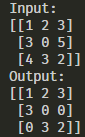


## پاسخ آ)

تابع ریاضی زیر:

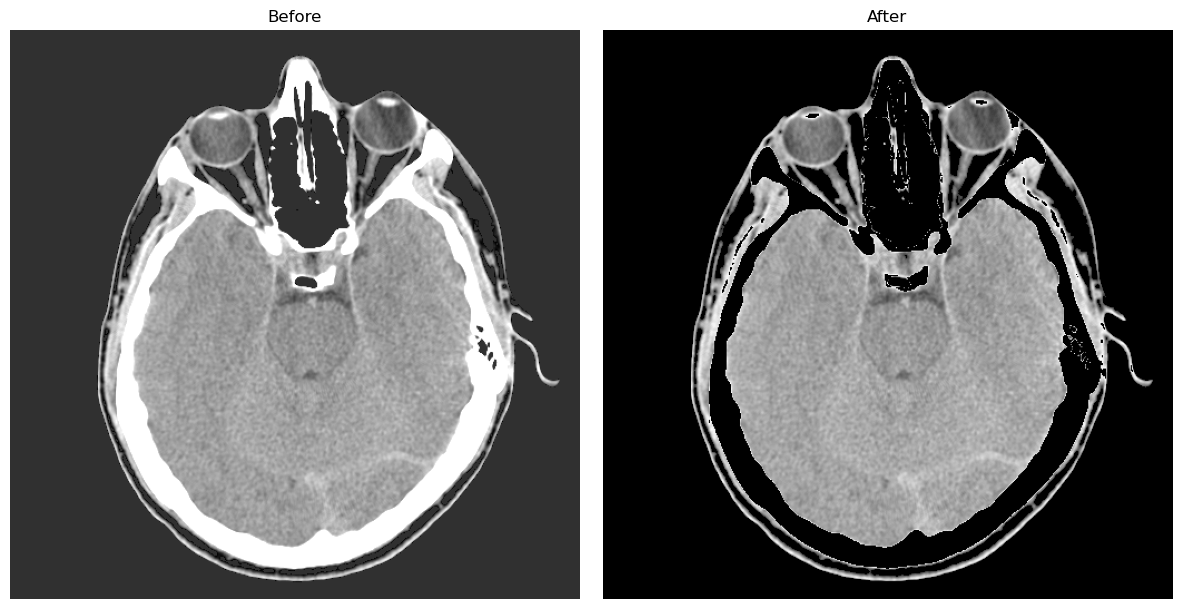
## پاسخ ب)

تابع را نوشتم. برای نوشتن این تابع به کمک یک حلقه for، تابع ریاضی بالا را روی هر پیکسل اعمال می‌شود و خروجی را برمی‌گرداند. برای مثال به ازای A=1,B=3 و ورودی پایین، خروجی پایین را خواهیم داشت:



## پاسخ ج)

با اعمال تابع بالا بر روی تصویر ورودی و به ازای A=55,B=250، خروجی زیر را خواهیم داشت:



## پاسخ د)

نمودار تابع تبدیل بدست آمده را نیز برای یک تصویر 8 بیتی بصورت زیر رسم کردم:

