در بخش اول (Part A) ابتدا تصویر از ورودی خوانده شده است

```
image = im2double(imread("image2.tif"));
```

سپس توسط تابع addNoiseToImage نویز پریودیک سینوسی به تصویر اعمال شده است.

```
noisyImage = addNoiseToImage(image);
```

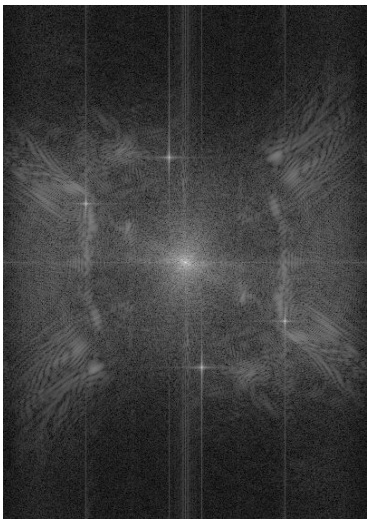
سپس طیف (Spectrum) تصویر نویزی در تبدیل فوریه به دست آمده است.

```
dftOfImage = fft2(concentrateSpectrum(noisyImage));
spectrumOfImage = abs(dftOfImage);
```

حال برای اینکه طیف را برای ویرایش کاربر ذخیره کنیم باید اولا کاری کنیم که طیف قابل نمایش باشد که از تابع log استفاده میکنیم که به دلیل امکان وجود عدد 0 در Spectrum ابتدا آن را با عدد 1 جمع کرده ایم. بعد از گرفتن لگاریتم باید کاری کنیم که کل اعداد به بازه ی 0 تا 255 منتقل شوند که به این دلیل از تابع normalizeMatrix استفاده شده است و برای دخیره تصویر هم قالب صحیح بدون علامت 8 بیتی استفاده شده است.

```
specImage = uint8(normalizeMatrix(log10(spectrumOfImage + 1) , 0 , 255));
```

بعد از این کار نوبت به نمایش تصویر اصلی و تصویر نویزی و طیف میرسد که توسط تابع imshow انجام شده است.



سپس در تابع main منتظر خواهیم شد تا کاربر تصویر طیف را ویرایش کرده و نویز را حذف نماید و کلید Enter را فشار دهد که به کار ادامه دهیم.

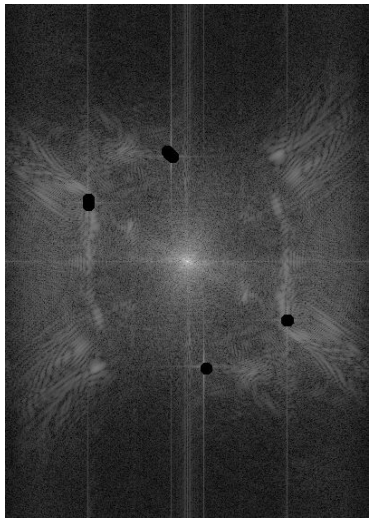
در بخش دوم(Part B) طیف ویرایش شده را میخوانیم.

```
editedSpectrumImage = double(imread("spectrumImage.tif"));
```

سپس طیف را به بازه ی اصلی خود برمیگردانیم . با توجه به اینکه ما طیف را بعلاوه 1 کرده و سپس لگاریتم گرفتیم و بعد از آن مقادیر را به بازه ی 0 تا 255 scale کردیم اکنون دقیقا برعکس عمل میکنیم . یعنی مقادیر را به بازه ی لگاریتمی مربوطه انتقال میدهیم و سپس 10 را به آن توان می رسانیم تا به بازه ی اصلی خود بازگردند.

```
enhancedSpec = 10 .^ normalizeMatrix(editedSpectrumImage ,
min(min(log10(spectrumOfImage+1))) , max(max(log10(spectrumOfImage+1))));
```

پس از اینکه طیف مجددا به اندازه ی کافی بزرگ شد آن را در فاز تصویر نویزی ضرب میکنیم و به محیط spatial بازمیگردانیم تا تصویر را بازسازی کنیم.

```
resultImage =
concentrateSpectrum(ifft2(enhancedSpec.*exp(1i*angle(dftOfImage))));
```

در نهایت همه ی تصاویر نمایش داده میشوند.