برنامه ای بنویسید که فریم‌های یک ویدئوی خام YUV را خوانده و مؤلفه Y فریم نخست را به صورت intra و مؤلفه Y فریم‌های دیگر را به صورت inter کدگذاری کند و بیت‌های حاصل را در فایل خروجی *outfile* ذخیره نماید. در این برنامه از برنامه‌های پیشین استفاده نمایید.

نکته۱: در فریم **اینترا** ، هر بلوک ۸ در ۸ پیکسلی پس از dct2، کوانتیزه می‌شود و بعد از زیگزاگ اسکن، Last Run Level هر بلوک به کمک دستور نکته۴ به بیت‌های هافمن تبدیل شده و در بیت‌ها فایل خروجی ذخیره می‌شود.

نکته۲: در فریم‌های **اینتر**، بایستی در ابتدا هر ماکروبلوک ۱۶ در ۱۶ پیکسلی از فریم فعلی به کمک فریم پیشین (البته فریم بازسازی شده پیشین) و MotionEstimation، تخمین زده و پیش بینی شود و بردارهای حرکت نیز به کمک دستور نکته۵ به بیت‌های هافمن تبدیل و در فایل خروجی ذخیره گردد. پس از پیش بینی فریم، ماکروبلوک خطا (یعنی اختلاف ماکروبلوک اصلی و ماکروبلوک پیش بینی) به ۴ بلوک ۸ در ۸ پیکسلی تقسیم می‌شود و هر بلوک بعد از dct2، با ضریب Q=16 کوانتیزه شده و مقادیر پس از زیگزاگ اسکن، دنباله Last Run Level تولید می‌شود. ماتریس Last Run Level هر بلوک جداگانه به کمک دستور هافمن نکته۴ به بیت‌های هافمن تبدیل می‌شود و در فایل خروجی ذخیره می‌شود.

نکته۳: برای خواندن مؤلفه Y فریم‌های یک فایل ویدئویی YUV، به صورت زیر عمل کنید:

```
input_file = fopen( 'Filename.yuv' , 'r' );
Y_frame1 = fread( input_file , [176 144] ) ' ;
fread( input_file , [88 72] ) ; fread( input_file , [88 72] ) ;
Y_frame2 = fread( input_file , [176 144] ) ' ;
fread( input_file , [88 72] ) ; fread( input_file , [88 72] ) ;
Y_frame3 = fread( input_file , [176 144] ) ' ;
fread( input_file , [88 72] ) ; fread( input_file , [88 72] ) ;
• • •
```

نکته۴: تابع **HuffCoding** ماتریس Last-Run-Level یک بلوک را دریافت می‌کند و بیت‌های هافمن نظیر این ماتریس را تولید می‌کند:

```
bits = HuffCoding( LRL );
```

به کمک دستور زیر، می‌توان بیت‌های تولید شده توسط **HuffCoding** را در فایلی با شناسه *outfile* ذخیره کرد:

```
fwrite( outfile , bits , 'ubit1' );
```

تابع **HuffCoding** و **fwrite** را به برنامه اضافه کنید تا بیت‌های حاصل از کدگذاری، در فایل خروجی ذخیره شود.

نکته۵: تابع **HuffCodingMV** دو بردار *mvr* و *mvc* حاصل از Motion Estimation را دریافت و بیت‌های هافمن نظیر را تولید می‌نماید:

```
bits = HuffCodingMV( mvr , mvc );
```

این تابع و تابع **fwrite** را در برنامه قرار دهید تا بیت‌های هافمن بردارهای حرکت در فایل خروجی (*outfile*) ذخیره شوند.

نکته۶: متغیر *outfile*، شناسه فایلی مانند EncodedVideo.mpeg است که بیت‌های حاصل از کدگذاری در آن ذخیره می‌شود. این متغیر بایستی به کمک دستور زیر و تنها در ابتدای فایل ایجاد شود:

```
outfile = fopen( 'EncodedVideo.mpeg' , 'w' );
```

و همچنین در انتهای برنامه هم به کمک دستور زیر بسته شود:

```
outfile = fclose( outfile );
```