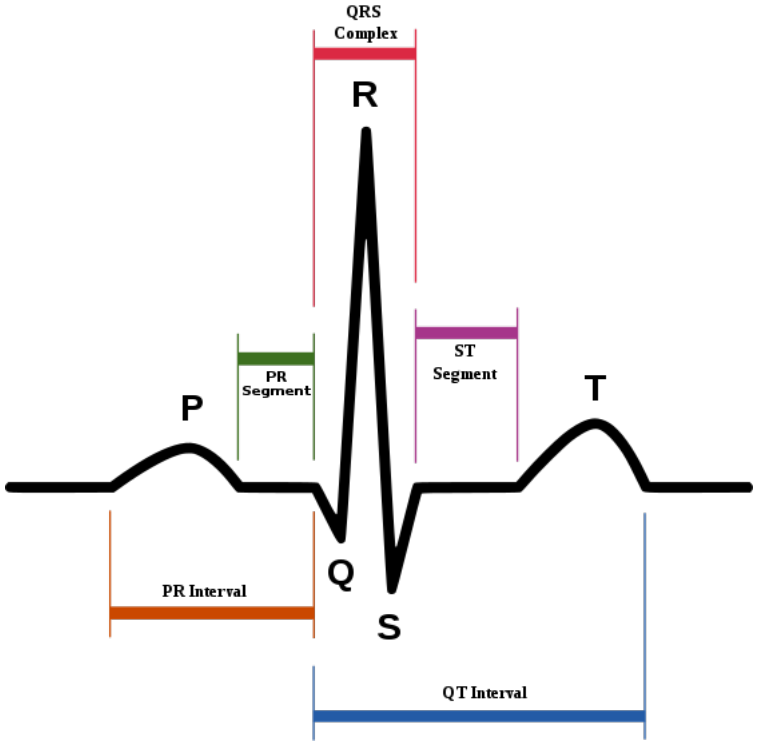
ECG Analysis

1)Obtain the data from a binary record file.

2)Visualize the obtained data.

ECG


QRS
Complex
R
PR
Segment
ST
Segment
P
T
Q
S
PR Interval
QT Interval

Detailed description about the record.

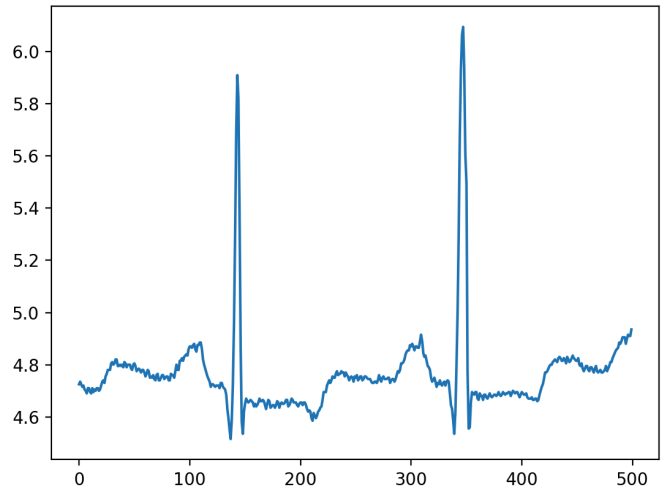
```
In [16]: record=wfdb.rdrecord('/Users/chaitanya/Downloads/ShubamData/qt-database-1.0.0/sel100',channels=[0])
         record.__dict__
```

```
Out[16]: {'record_name': 'sel100',
          'n_sig': 1,
          'fs': 250,
          'counter_freq': 360.0,
          'base_counter': None,
          'sig_len': 225000,
          'base_time': None,
          'base_date': None,
          'comments': ['69 M 1085 1629 x1',
           'Aldomet, Inderal',
           'Produced by xform from record 100, beginning at 7:00.000'],
          'sig_name': ['MLII'],
          'p_signal': array([[4.725],
                 [4.735],
                 [4.725],
                 ...,
                 [4.695],
                 [4.68 ],
                 [4.68 ]]),
          'd_signal': None,
          'e_p_signal': None,
          'e_d_signal': None,
          'file_name': ['sel100.dat'],
          'fmt': ['212'],
          'samps_per_frame': [1],
          'skew': [None],
          'byte_offset': [None],
          'adc_gain': [200.0],
          'baseline': [0],
          'units': ['mV'],
          'adc_res': [11],
          'adc_zero': [1024],
          'init_value': [945],
          'checksum': [-13873],
          'block_size': [0]}
```

Visualizing the signal.

In [371]:
```
signal_slice=np.ndarray.flatten(record.p_signal[0:500])
peak_index=peakutils.indexes(signal_slice,thres=0.3,min_dist=100)
fig,ax=plt.subplots()
ax.plot(signal_slice)

# for peak in peak_index:
#     ax.axvline(x=peak,color='r')
len(record.p_signal)
```

Figure 49

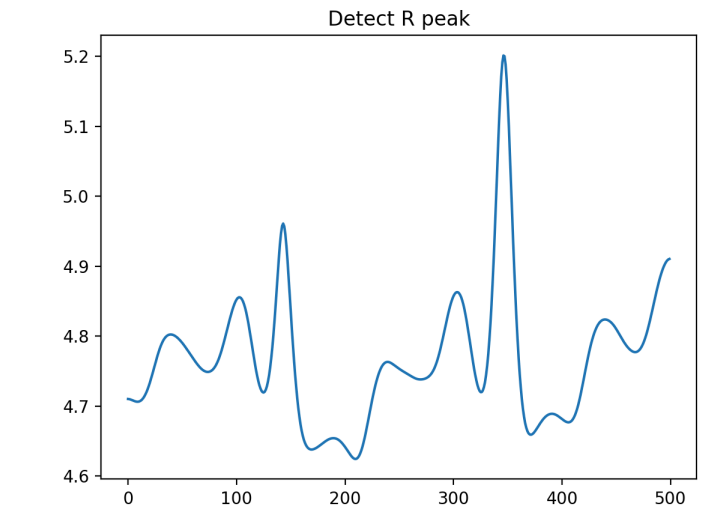


Out[371]: 225000

Smoothing the signal.

In [398]:
```
smoothed_signal=signal.cspline1d(signal_slice,lamb=1000)
r_peaks=peakutils.indexes(smoothed_signal,thres=0.5,min_dist=0.1)

fig,ax=plt.subplots()
ax.set_title('Detect R peak')
ax.plot(smoothed_signal)

# for peak in r_peaks:
#     ax.axvline(x=peak,color='r')
```

Figure 62

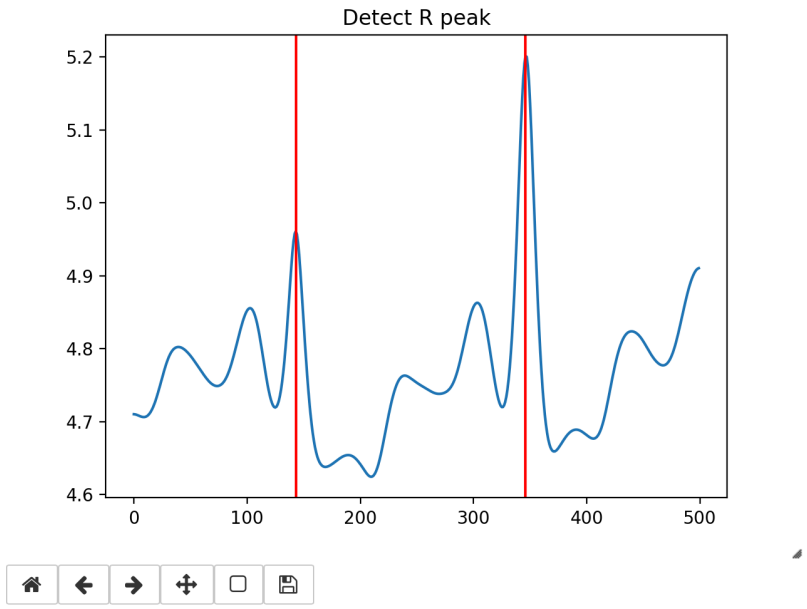


Out[398]: [<matplotlib.lines.Line2D at 0x1c594b1b10>]

Detecting R-peak

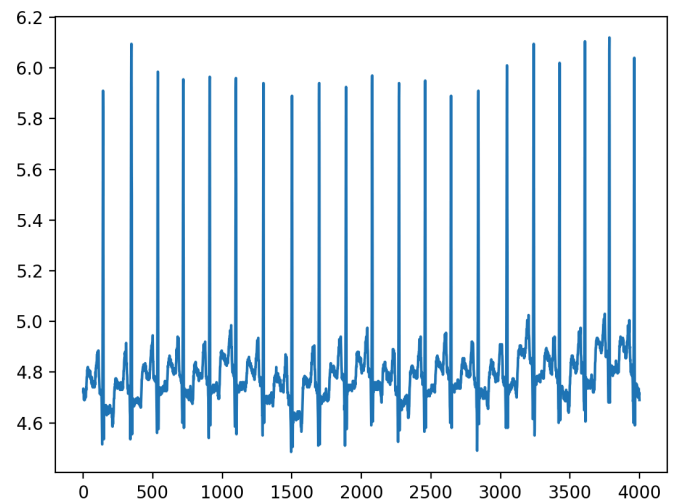
In [399]:
```
smoothed_signal=signal.cspline1d(signal_slice,lamb=1000)
r_peaks=peakutils.indexes(smoothed_signal,thres=0.5,min_dist=0.1)

fig,ax=plt.subplots()
ax.set_title('Detect R peak')
ax.plot(smoothed_signal)

for peak in r_peaks:
    ax.axvline(x=peak,color='r')
```

Figure 62



In [372]:
```
signal_slice=np.ndarray.flatten(record.p_signal[0:4000])
peak_index=peakutils.indexes(signal_slice,thres=0.3,min_dist=100)
fig,ax=plt.subplots()
ax.plot(signal_slice)

# for peak in peak_index:
#     ax.axvline(x=peak,color='r')
len(record.p_signal)
```

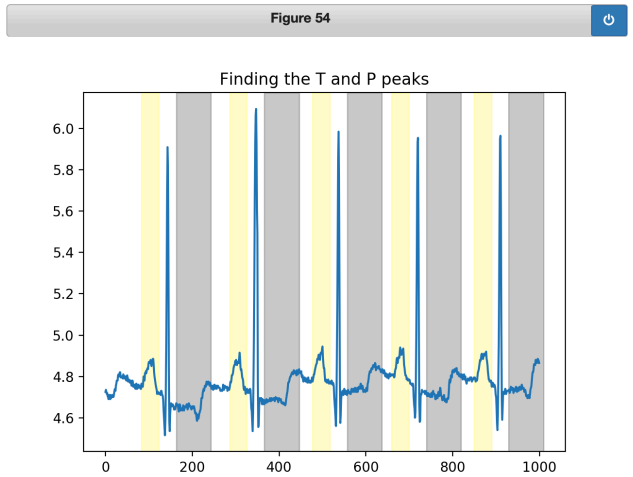
Figure 49



Out[372]: 225000

Detecting P and T peaks.

In [379]:
```
fig,ax=plt.subplots()

p_min=-20
p_max=-60


for peak in r_peaks:
    ax.axvspan(peak+p_max,peak+p_min,alpha=0.2,color='yellow')

t_min=20
t_max=100



for peak in r_peaks:
    ax.axvspan(peak+t_max,peak+t_min,alpha=0.2,color='black')

ax.set_title('Finding the T and P peaks')
ax.plot(signal_slice)
print(ax)
```




```
AxesSubplot(0.125,0.11;0.775x0.77)
```

Tried getting a Point cloud.

In [383]:
```
P=[]
Q=[]
R=[]
S=[]
T=[]
for i in range(len(r_peaks)):
    P.append([np.max(signal_slice[pm[i]:pmin[i]]),i])
    Q.append([np.min(signal_slice[pmin[i]:r_peaks[i]]),i])
    R.append([np.max(signal_slice[r_peaks[i]]),i])
    S.append([np.min(signal_slice[r_peaks[i]:tmin[i]]),i])
    T.append([np.max(signal_slice[tmin[i]:tm[i]]),i])
```

Point cloud

```
In [384]: P
Out[384]: [[4.885, 0], [4.915, 1], [4.945, 2], [4.94, 3], [4.92, 4]]

In [385]: Q
Out[385]: [[4.515, 0], [4.535, 1], [4.56, 2], [4.6, 3], [4.54, 4]]

In [386]: R
Out[386]: [[5.91, 0], [6.065, 1], [5.985, 2], [5.935, 3], [5.95, 4]]

In [387]: S
Out[387]: [[4.535, 0], [4.555, 1], [4.575, 2], [4.58, 3], [4.59, 4]]

In [388]: T
Out[388]: [[4.775, 0], [4.83, 1], [4.865, 2], [4.84, 3], [4.885, 4]]

In [ ]:

In [392]: fig,ax=plt.subplots()
          ax.plot(P,T)
```

Figure 56