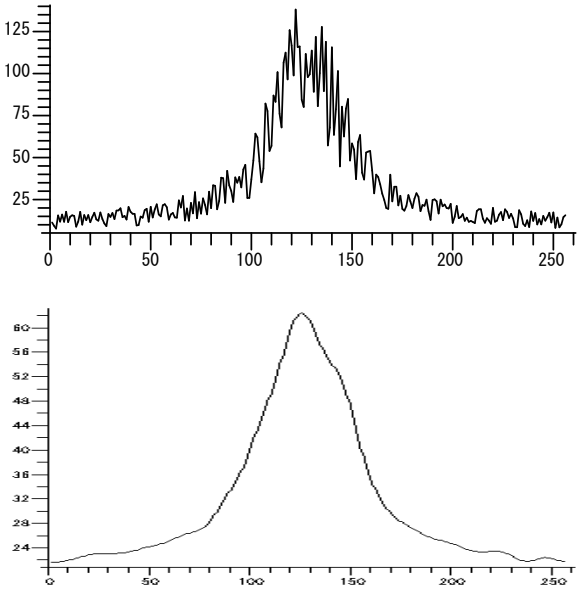
ーFFTー

課題

下図の上段パネルのようなデータをフーリエ変換し，三角フィルタによってノイズを除去し，下段パネルのようにせよ．



データは次のようにして作成・表示した．

```
> f1:=t->subs({a=10,b=40000,c=380,d=128},a+b/(c+(t-d)^2)):
  T:=[seq(f1(i)*(0.6+0.8*evalf(rand()/10^12)),i=1..256)]:
  with(plots):listplot(T);
```

課題の背景

信号処理においては，フーリエ変換を用いたデータのフィルタリング

$$X(f)=\int_{-\infty}^{\infty} x(t)\exp(2\pi I f t)dt$$

が多用される．これは，高速フーリエ変換(*Fast Fourier Transformation*:略して*FFT*)アルゴリズムの発明によって計算機での処理が容易となったためである．*FFT*によって，時間経過と共に変化する時系列測定データ$x(t)$に含まれるさまざまな周波数成分について，何*Hz*の周波数成分がどの程度の割合でその時系列データの中に含まれているのかを明らかにできる．時系列データの中のノイズはδ関数に似ており，そのフーリエ変換は全周波数成分を一様に含んでいる．一方，測定したいデータは一般に，時間に対してゆっくり変化する量である．測定によって得られたデータの時間軸は連続ではなくとびとびの値をとるので，離散フーリエ変換を取ることになる．この例では時間データは*256*チャンネルのデータ．高周波成分（=ノイズ成分）を取り除くために，データに対して窓関数（低域通過フィルタ）をかけてフーリエ逆変換

$$x(t)=\int_{-\infty}^{\infty} X(f)\exp(2\pi I f t)dt$$

することによってスムーズなカーブが得られる．この他の低域通過フィルタには，方形窓，修正方形窓，フォンハン窓（２乗余弦窓），ハミング窓などさまざまなものがある．

Mapleの関連コマンド

実際のFFTを少ないデータでしめす．まずデータを用意する．

```
> Rdata:=Array([1,2,3,4,5,4,3,2]);
  Idata:=Array(1..8,0);
```

$$Rdata := \left[\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 4 & 3 & 2 \end{array}\right]$$

$$Idata := \left[\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \qquad (1.3.1)$$

これにFFTをかけます．

```
> FFT(3,Rdata,Idata);
```

$$8 \qquad (1.3.2)$$

これだけで終わり．ここでFFTの第一引数の3はデータの個数が2^3であることを示している．フーリエ変換された後のデータ構造を見てみよう．

```
> print(Rdata):print(Idata):
```

$$\left[\begin{array}{l}[24, -6.828427123, 0., -1.171572877, 0, -1.171572877, 0.,\\ \quad -6.828427123]\end{array}\right]$$

$$\left[\begin{array}{cccccccc} 0 & 1.10^{-9} & 0. & 1.10^{-9} & 0 & -1.10^{-9} & 0. & -1.10^{-9} \end{array}\right] \qquad (1.3.3)$$

逆変換とその結果は以下の通りである．．

```
> iFFT(3,Rdata,Idata);
  print(Rdata):print(Idata):
```

$$8$$

$$\left[\begin{array}{l}[1.000000000, 2.000000001, 3.000000000, 3.999999999,\end{array}\right.$$

5.000000000, 3.999999999, 3.000000000, 2.000000001]

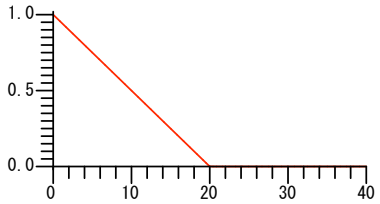
[0., −2.500000000 10^{-10}, 0., −2.500000000 10^{-10}, 0., (1.3.4)

2.500000000 10^{-10}, 0., 2.500000000 10^{-10}]

三角フィルター関数は以下のようにして作られる.

```
> filter:=x->piecewise(x>=0 and x<=20, (1-x/20));
  plot(filter(x),x=0..40);
```

$$\text{filter} := x \rightarrow \text{piecewise}\left(0 \le x \text{ and } x \le 20, 1 - \frac{1}{20}x\right)$$



>

解法のヒント

1 256個のデータをRdataに読み込め.

2 FFTをかけよ.

3 FFTをかけた後のデータの強度(Rdata[i]^2+Idata[i]^2)をlogplotせよ.

4 FFTをかけた後のデータにfilter関数を掛けよ.

5 フーリエ逆変換してその結果を表示せよ.