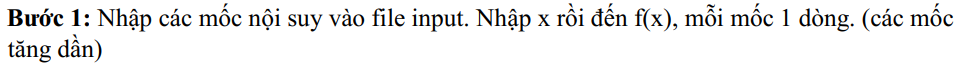
* nhập input : 

Input :

* Các mốc nội suy & giá trị nội suy theo thứ tự tiến (lùi)
* X0 giá trị cần xấp xỉ
* N : số lượng các mốc cần kết nạp

Output :

* Hệ số đa thức nội suy tiến (lùi)
* Giá trị xấp xỉ tại X0

Thuật toán chính

1. Input() : nhập các mốc nội suy từ file và kiểm tra điều kiện
2. Gói chọn điểm () : Chọn ra N điểm gần X0 nhất
   * Input() : X0,N,các vecto mốc nội suy x,y
   * B1 : kiểm tra
     + if N > len(x) => báo lỗi dừng chương trình
     + else
       - Gán h = x[1] – x[0] // Khoảng cách đều
       - t =(int) // t sẽ được làm tròn
       - Gán biến left = min(len(x) – num, max(0,t+1-int(num/2)) // Xử lý các trường hợp điểm cần xấp xỉ nằm sát ngoài biên
       - Right = left + num + 1
       - Khởi tạo 2 mảng mới x1,y1 để lưu giá trị nội suy cần kết nạp
       - X1=x[left : right +1]
       - Y1=y[left : right +1]

* Trả về x1,y1

1. Gói tính giá trị bằng lược đồ Hoocne

* trong phần hoocne

1. Gói xây dựng bảng sai phân các cấp
   * Input : Ma trận BSP,x,y ( các mốc nội suy ) , i ( n= len(x)-1)
   * Output : Vecto hàng của bảng sai phân
     + Khởi tạo ma trận không SP\_row\_i(len(x))
     + SP\_row\_i[0] = y[i]
     + For j =1 to i
       - SP\_row\_i[j] =-BSP[j-1] - SP\_row\_i[j-1]
     + Return SP\_row\_i
2. BTT : Kết nạp mốc x[i] vào bảng tính tích theo lược đồ Hoocne
   * Input : Ma trận không BTT, x( vecto cột mốc nội suy đã được chọn ),i ( i=len(x)-1)
   * Output : vecto cột của bảng tính tích hệ số
     + Khởi tạo mảng không TT\_row\_i ( len(x))
     + Gán giá trị TT\_row\_i[i]=1
     + For j=i , j <1 do
       - TT\_row\_i[i]= BTT[j-1]-i\*BTT[j]
     + If i!=0 then
       - TT\_row\_i[i]= -i\*BTT[0]

Return TT\_row\_i[i]

1. Gói fx

* Input : BSP,BTT,I,f
* Output : Hệ số của đa thức nội suy
  + Begin
    - For j=1 --> do
      * f[j]= f[j]+
  + Return f

1. Gói NS\_Newton\_Lui :
   * sử dụng các gói :
     + tính BSP
     + BTT
     + Fx