1. a. Error - 3 arguments expected; only integer arguments allowed
   b. tiS

2. The list being traversed is being modified alongside. Elements are being deleted while traversal, so the loop control variable would skip elements in between, this may result in elements that should have been deleted stay in the final list. Also, the loop shall run upto the initial length of the list, which would be reduced, so index out of range error is expected.

   The correct way to do this would be through list comprehension as follows.

```
def func(my_list):
  my_list = [i for i in my_list if i<5]
  print(my_list)
```

3. (i)

```
count=0
for i in range(len(s)//2):
   if s[i]==s[-1-i]:
      count +=1
return count
```

   (ii)

```
sum = 0 # Intermedite accumulator
# Mutability requires loops over positions
for pos in range(len(lst)):
         sum = sum + lst[pos]
         lst[pos] = sum
# No return value
```

   (iii)

```
def is_prime(x):
  if x==1 or x==2 or x==3:
    return True
  for i in range(2,int(x**0.5)+1):
    if(x%i==0):
      return False
  return True

def func(a,b):
  lst = []
  for i in range(a,b+1):
    if not(is_prime(i)):
      lst.append(i)
```

(iv)

```
import math

def collapse(ragged):
  for i in range(len(ragged)):
    l = len(ragged[i])
    if l==0:
      ragged[i] = 0.0
    else:
      ragged[i] = math.fsum(ragged[i])/l

  print(ragged)
```