6.00 Introduction to Computer Science and Programming
Fall 2008

Lecture 10 handout
6.00 Fall Term 2008

```
def merge(left,right):
  """Assumes left and right are sorted lists.
Returns a new sorted list containing the same elements
as (left + right) would contain."""
  result = []
  i,j = 0, 0
  while i < len(left) and j < len(right):
      if left[i] <= right[j]:
         result.append(left[i])
         i = i + 1
      else:
         result.append(right[j])
         j = j + 1
  while (i < len(left)):
      result.append(left[i])
      i = i + 1
  while (j < len(right)):
      result.append(right[j])
      j = j + 1
  return result

def mergesort(L):
   """Returns a new sorted list with the same elements as L"""
   print L
   if len(L) < 2:
      return L[:]
   else:
      middle = len(L) / 2
      left = mergesort(L[:middle])
      right = mergesort(L[middle:])
      together = merge(left,right)
      print 'merged', together
      return together


def create(smallest, largest):
    intSet = []
    for i in range(smallest, largest+1): intSet.append(None)
    return intSet

def insert(intSet, e):
    intSet[e] = 1

def member(intSet, e):
    return intSet[e] == 1




def hashChar(c):
    # c is a char
    # function returns a different integer in the range 0-255
    # for each possible value of c
    return ord(c)
```

```
def cSetCreate():
    cSet = []
    for i in range(0, 255): cSet.append(None)
    return cSet

def cSetInsert(cSet, e):
    cSet[hashChar(e)] = 1

def cSetMember(cSet, e):
    return cSet[hashChar(e)] == 1


def readFloat(requestMsg, errorMsg):
    while True:
        val = raw_input(requestMsg)
        try:
            val = float(val)
            return val
        except:
            print(errorMsg)

#print readFloat('Enter float: ', 'Not a float.')

def readVal(valType, requestMsg, errorMsg):
    while True:
        val = raw_input(requestMsg)
        try:
            val = valType(val)
            return val
        except:
            print(errorMsg)

# print readVal(int, 'Enter int: ', 'Not an int.')
```