
```
# CMPT 145 Course material
# Copyright (c) 2017-2021 Michael C Horsch
# All rights reserved.
#
# This document contains resources for homework assigned to students of
# CMPT 145 and shall not be distributed without permission. Posting this
# file to a public or private website, or providing this file to a person
# not registered in CMPT 145, constitutes Academic Misconduct, according
# to the University of Saskatchewan Policy on Academic Misconduct.
#
# Synopsis:
#   Assignment 7 solutions
```

```
def recsumlist(alist):
    """
    Purpose:
        Return the sum of the values in the list
    Preconditions:
        :param alist: a list containing numbers
    Return:
        :return: the sum of the values in the list
    """
    if len(alist) == 0:
        return 0
    else:
        return alist[0] + recsumlist(alist[1:])

def recmemberlist(value, alist):
    """
    Purpose:
        Check if the given value appears in the given list.
    Preconditions:
        :param value: a value
        :param alist: a list of values
    Return:
        :return: True if the value appears in the list, False otherwise
    """
    if len(alist) == 0:
        return False
    elif alist[0] == value:
        return True
    else:
        return recmemberlist(value, alist[1:])

def reccountlist(value, alist):
    """
    Purpose:
        Count the number of times the given value appears in the given list.
    Preconditions:
        :param value: a value
        :param alist: a list of values
    Return:
        :return: the number of times the value appears in the list
    """
    if len(alist) == 0:
        return 0
```

```

elif alist[0] == value:
    return 1 + reccountlist(value, alist[1:])
else:
    return reccountlist(value, alist[1:])

def subst_str(t, r, s):
    """
    Purpose:
        Return a string that has the same characters in s, except that
        every occurrence of character t is replaced by character r.
    Preconditions
        :param t: the target character to replace
        :param r: the replacement character
        :param s: a string
    Return
        :return: a new string with t replaced by r in s
    """
    if len(s)==0:
        return ''
    elif s[0] == t:
        return r + subst_str(t, r, s[1:])
    else:
        return s[0] + subst_str(t, r, s[1:])

```