**CSC 148 Lab 7**

List, List comprehension, Sorting, Tuples.

**Total points 30pts**

Due: November 12, 2021

1. (5pts) Given a list of numbers, return a list where all adjacent elements have been reduced to a single element, so [1, 2, 2, 2, 3,3,4] returns [1, 2, 3, 4]. You may create a new list or modify the passed in list.

Hint: there are several solutions. But one is particularly elegant!

*def remove\_adjacent(num\_list):*

*# your code here*

2. (5pts) Use list comprehension to finish the following questions.

a) Given a list of words, return the list of words with word length less than 5.

*def less\_thanfive(word\_list):*

*# your code here*

b) Use the “filter” function to solve the above question.

Hint: you can define another function:

*def less\_thanfive\_filter(word):*

*# your code here*

3. (5pts) Can you write a function reverse\_list(word\_list) which takes in a list as an argument and returns a list with its elements reversed? **You can’t use the list methods reverse() for this question.**

**Example:** reverse\_list([‘a’, ‘b’, ‘c’, ‘d’]) [‘d’, ‘c’, ‘b’, ‘a’]

*def reverse\_list(input\_list):*

*#your code here*

The easiest solution is to use a new list. Can you think of a solution WITHOUT using a new list? **This is called in-place algorithm.** You can swap the element symmetrically.

4. (5pts) Write a function dot\_product(u, v) that takes two lists of numbers of the **same** length, and returns the sum of the products of the corresponding elements of each (the [dot\_product](http://en.wikipedia.org/wiki/Dot_product)).

dot\_product([1, 1], [1, 1]) returns 2

dot\_product([1, 2], [1, 4]) returns 9

dot\_product([1, 2, 1], [1, 4, 3]) returns 12

5. (5pts)

Write a program that creates a two-dimensional list called heights that are based on user inputs.

Ask the user to input a name and height in inches until they input “done” as their name.

Save the information they entered into a 2D list like the following.

heights = [["Noelle", 61], ["Ava", 70], ["Sam", 67], ["Mia", 64]]

Return the 2D list.

6. (5pts)

**Blocks of Stock.**A block of stock has a number of attributes, including a purchase date, a purchase price, a number of shares, and a ticker symbol, and a current price. We can record these pieces of information in a tuple for each block of stock and do a number of simple operations on the blocks.

Assume that we have the following portfolio.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Purchase Date** | **Purchase Price** | **Shares** | **Symbol** | **Current Price** |
| 25 Jan 2001 | 43.50 | 25 | CAT | 92.45 |
| 25 Jan 2001 | 42.80 | 50 | DD | 51.19 |
| 25 Jan 2001 | 42.10 | 75 | EK | 34.87 |
| 25 Jan 2001 | 37.58 | 100 | GM | 37.58 |

We can represent each block of stock as a 5-tuple with purchase date, purchase price, shares, ticker symbol and current price.

portfolio= [ ( "25-Jan-2001", 43.50, 25, 'CAT', 92.45 ), ( "25-Jan-2001", 42.80, 50, 'DD', 51.19 ), ( "25-Jan-2001", 42.10, 75, 'EK', 34.87 ), ( "25-Jan-2001", 37.58, 100, 'GM', 37.58 ) ]

1. Develop a function that examines each block, multiplies shares by purchase price and determines the total purchase price of the portfolio.
2. Develop a second function that examines each block, multiplies shares by purchase price and shares by current price to determine the total amount gained or lost.