CPSC 1301, Computer Science I Programming Exercise

Week 04, List Play

1 Exercise Requirements

This activity consists of four programming exercises. The following exercises are open book and open note. You are free to use any written documentation you wish. However, these are individual exercises, and you cannot consult with each other in writing your programs. Name your program `ListPlay_lastname.py`.

This programming exercise has four parts consisting of four requirements. The grade for each requirement is indicated, for a maximum of 100 points. At a minimum, your program must compile successfully and run.

For this programming exercise, you have five test lists of type integer, shown below. Write a program that performs the following tasks, listed below. **Do not** use convenience functions. (You may use the `len` function.) Write the code by hand.

List A: [0, 2, 4, 6, 8, 10]

List B: [1, 3, 5, 7, 9]

List C: [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 9]

List D: ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

List E: ['Y', 'o', 'u', 'r', ' ', 'N', 'a', 'm', 'e']

2 Exercise Requirements

Printing the lists: 70 points Write a method that accepts a list parameter and prints each list. See the output below.

Reversing lists: 80 points Write a method accepts a list parameter and prints each list backwards. See the output below.

Rotating lists: 90 points Lists can be rotated to the right or to the left by any number of places. Rotating a list to the right by two places puts the first element in position three, the second element in position four, and so on, with the last element in position two and the next to last element in position one. List A rotated to the right by two places results in [8, 10, 0, 2, 4, 6]. Likewise, rotating an list to the left places the first elements at the end. List B rotated to the left by two places results in [5, 7, 9, 1, 3].

Write a method that accepts three parameters, a direction (right or left), a number of places, and an list, which prints the appropriate rotation. Call the method on all three lists. Rotate list A two places to the left. Rotate list B two places to the right. Rotate list C four places to the left. Rotate lists D and E to the right and left, one in each direction, our choice. See the output below.

Sorting lists: 100 points Write a method that takes an unsorted integer list and prints a sorted list. Use Array C as your test list. Do you recognize this list of numbers? It also works for alphabetical characters. Sort your name, List E.

HINT: How do you sort a deck of cards? The simplest strategy is to go through the deck and if the "next" card is "smaller" than the current card, swap them. Then, repeat the process until all cards are in order. You should try to implement this without any outside help, but if you need just a hint, check out *bubble sort.*

3 Starter Template

```
#!python
# Name: ListPlay.py
# Author: Your Name
# Date: current date
# Purpose: playing with lists

def hello():
    print("Hello from 'ListPlay.py'")

def print_list(li):
    pass

def print_list_reversed(li):
    pass

def print_list_rotated(li, direction, place):
    pass

def print_bubble_sort(li):
    pass

#main function executes the defined functions
if __name__ == '__main__':
    hello()

    x1 = [0,2,4,6,8]
    x2 = [1,3,5,7,9]
    x3 = [3,1,4,1,5,9,2,6,5,3,5]
    x4 = "abcdefghij"
    x5 = "Your Name"

    print("printing the lists")
    print_list(x1)
    print_list(x2)
    print_list(x3)
    print_list(x4)
    print_list(x5)

    print("printing the lists in reverse order")
    print_list_reversed(x1)
    print_list_reversed(x2)
    print_list_reversed(x3)
    print_list_reversed(x4)
    print_list_reversed(x5)

    print("printing the rotated lists")
    print_list_rotated(x1, "L", 2)
    print_list_rotated(x2, "R", 3)
    print_list_rotated(x3, "L", 4)
    print_list_rotated(x4, "R", 5)
    print_list_rotated(x5, "L", 6)

    print("printing the sorted list")
    print_bubble_sort(x3)
    print_bubble_sort(list(x5)) #casting the string to a list
```

4 Sample Output

```
Hello from 'ListPlay.py'
printing the lists
0  2  4  6  8
1  3  5  7  9
```

```
3  1  4  1  5  9  2  6  5  3  5
a  b  c  d  e  f  g  h  i  j
Y  o  u  r     N  a  m  e
printing the lists in reverse order
8  6  4  2  0
9  7  5  3  1
5  3  5  6  2  9  5  1  4  1  3
j  i  h  g  f  e  d  c  b  a
e  m  a  N     r  u  o  Y
printing the rotated lists
4  6  8  0  2
5  7  9  1  3
5  9  2  6  5  3  5  3  1  4  1
f  g  h  i  j  a  b  c  d  e
a  m  e  Y  o  u  r     N
printing the sorted list
1  1  2  3  3  4  5  5  5  6  9
   N  Y  a  e  m  o  r  u
```