Name: ________________________________ Date: ________________

Unit 2 Test: Abstractions and Lists

Score: ____ + _____ = ______ / 24

bonus

1. D
2. C
3. A
4. B
5. D
6. A
7. A
8. C
9. B
10. A
11. B
12. C

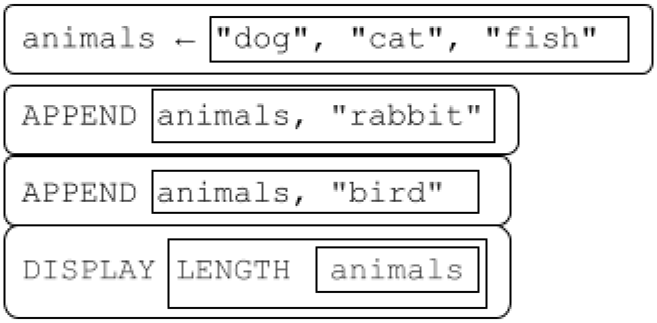
BONUS: Please use the given answer boxes on the last page.

Unit 2 Test: Abstractions and Lists

Name: ________________________________ **Date:** ________________

1. The list `wordList` contains a list of 10 string values. Which of the following is a valid index for the list?
   A. `-1`
   B. `"Hello"`
   C. `2.5`
   D. `4`

2. Consider the following code segment:



```
animals ← "dog", "cat", "fish"
APPEND animals, "rabbit"
APPEND animals, "bird"
DISPLAY LENGTH animals
```

   After the code segment is executed, what is the output?

   A. `"dog", "cat", "fish", "rabbit", "bird"`
   B. `3`
   C. `5`
   D. `4`

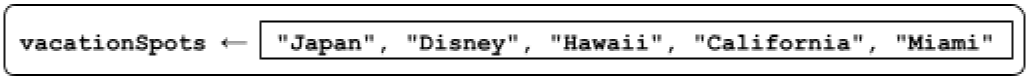
3. Consider the following code segment:

```
numberList ← [7, 5, 1, 3]
X ← numberList[1] + numberList[3]
DISPLAY(X)
```

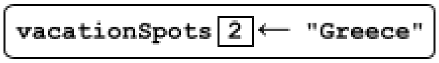
   What will be displayed after executing the code segment?

   A. 8
   B. 4
   C. 10
   D. 13

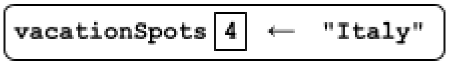
4. The following code segment uses lists to store and update the top 5 vacation destinations:



```
vacationSpots ← "Japan", "Disney", "Hawaii", "California", "Miami"
```



```
vacationSpots 2 ← "Greece"
```



```
vacationSpots 4 ← "Italy"
```

After running that code, what does the `vacationSpots` list store?

A. "Japan", "Disney", "Hawaii", "California", "Miami", "Greece", "Italy"
B. "Japan", "Greece", "Hawaii", "Italy", "Miami"
C. "Japan", "Hawaii", "Miami", "Greece", "Italy"
D. "Japan", "Hawaii", "Miami"

5. This list represents the leading cars in a race, according to the car numbers:

```
raceCars ← [18, 2, 42, 10, 4, 1,  6, 3]
```

This code snippet updates the list:

```
tempCar ← raceCars[6]
raceCars[6] ← raceCars[5]
raceCars[5] ← tempCar
```

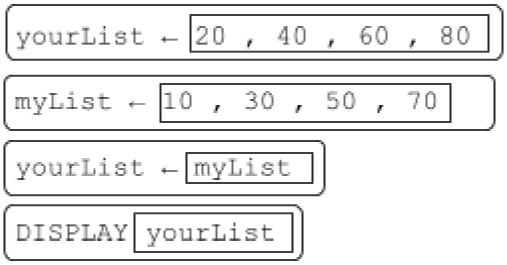
What does the `raceCars` variable store after that code runs?

A. 18, 2, 42, 10, 4, 6, 1, 3
B. 18, 2, 42, 10, 1, 1, 6, 3
C. 18, 2, 42, 10, 6, 4, 1, 3
D. 18, 2, 42, 10, 1, 4, 6, 3

6. Which of the following is a benefit of using a list as a data abstraction in a program?

A. Lists often allow their size to be easily updated to hold as many data values as needed.
B. Lists convert all elements to strings so that they can be inspected character-by-character.
C. Lists prevent duplicate data values from appearing in the list.
D. Lists are used to store all input data so that there is a running record of all user input.

7. Consider the following code segment. What will be displayed after executing the code segment?



```
yourList ← 20 , 40 , 60 , 80
myList ← 10 , 30 , 50 , 70
yourList ← myList
DISPLAY yourList
```

   A. `[10, 30, 50, 70]`
   B. `[20, 40, 60, 80]`
   C. `[10, 30, 50, 70, 20, 40, 60, 80]`
   D. `[20, 40, 60, 80, 10, 30, 50, 70]`

8. A local search website lets users create lists of their favorite restaurants. When the user first starts, the website runs this code to create an empty list:

```
localFavs ← []
```

The user can then insert and remove items from the list. Here's the code that was executed from one user's list making session:

```
APPEND(localFavs, "Spanky’s")
APPEND(localFavs, "Avo Taco")
APPEND(localFavs, "Panera Bread")
APPEND(localFavs, "Luigi’s")

INSERT(localFavs, 3, "Burger King")
REMOVE(localFavs, 2)
```

What does the `localFavs` variable store after that code runs?

   A. `"Spanky’s", "Avo Taco", "Panera Bread", "Luigi’s", "Burger King"`
   B. `"Spanky’s", "Burger King", "Avo Taco", "Panera Bread", "Luigi’s"`
   C. `"Spanky’s", "Burger King", "Panera Bread", "Luigi’s"`
   D. `"Spanky’s", "Burger King", "Luigi’s"`

9. A student is creating a fortune teller program. Which of the following would be an appropriate algorithm??
   A. Create a list of fortunes, ask the user to input a number, output the first fortune list item.
   B. Create a list of fortunes, ask the user to input a number, output the fortune list item at the number the user chose.
   C. Ask the user for input for fortunes, create a list of fortunes, output a random number.
   D. Create a list of fortunes, Ask the user for input for fortunes, output the first fortune list item

10. This code snippet stores and updates a list that represents files in a folder:

```
fileNames ← ["cow.mov", "dog.wav",  "cat.jpg", "bird.avi", "fly.gif"]

DISPLAY(fileNames[3])
INSERT(fileNames,  2, "goat.tif")
INSERT(fileNames, 6, "spider.html")

DISPLAY(fileNames[3]
```

What does this program output to the display?

A. `cat.jpg dog.wav`
B. `cat.jpg cat.jpg`
C. `bird.avi dog.wav`
D. `cat.jpg cow.mov`

11. Consider the following code segment.

```
firstList ← ["guitar", "drums", "bass"]
secondList ← ["flute", "violin"]
thirdList ← []
thirdList ← firstList
firstList ← secondList
secondList ← thirdList
```

What are the contents of `secondList` after the code segment is executed?

A. `[]`
B. `["guitar", "drums", "bass"]`
C. `["flute", "violin"]`
D. `["flute", "violin", "guitar", "drums", "bass"]`

12. A code segment is intended to transform the list `utensils` so that the last element of the list is moved to the beginning of the list. For example, if `utensils` initially contains `["fork", "spoon", "tongs", "spatula", "whisk"]`, it should contain `["whisk", "fork", "spoon", "tongs", "spatula"]` after executing the code segment.

Which of the following code segments transforms the list as intended?

A.
```
len ← LENGTH(utensils)
temp ← utensils[len]
REMOVE(utensils, len)
APPEND(utensils, temp)
```

B.
```
len ← LENGTH(utensils)
REMOVE(utensils, len)
temp ← utensils[len]
APPEND(utensils, temp)
```

C.
```
len ← LENGTH(utensils)
temp ← utensils[len]
REMOVE(utensils, len)
INSERT(utensils, 1, temp)
```

D.
```
len ← LENGTH(utensils)
REMOVE(utensils, len)
temp ← utensils[len]
INSERT(utensils, 1, temp
```

Unit 2 Test: Abstractions and Lists

BONUS: Match the following code segments to their corresponding explanations.
Write the letter choice in the box.

A. Takes an element from the list and assigns it to a variable.

B. Assigns a copy of one list to another list.

C. Creates an empty list.

D. Adds an element to a certain position in the list, where all elements after would get shifted to the right.

E. Assigns the value of a variable to an index on the list.

F. Adds an element to the end of a list.

G. Stores a value into a certain index in the list

H. Finds the length of the list

Answer □

Text:
```
INSERT(aList, i, value)
```
Block:
```
INSERT [aList, i, value]
```

Answer □

Text:
```
APPEND(aList, value)
```
Block:
```
APPEND [aList, value]
```

Answer □

Text:
```
aList ← []
```
Block:
```
aList ← [ ]
```

Answer □

Text:
```
aList ← bList
```
Block:
```
aList ← bList
```

Answer □

Text:
```
x ← aList[i]
```
Block:
```
x ← aList [i]
```

Answer □

Text:
```
LENGTH(aList)
```
Block:
```
LENGTH [aList]
```

Answer □

Text:
```
aList[i] ← x
```
Block:
```
aList [i] ← x
```

Answer □

Text:
```
aList[i] ← aList[j]
```
Block:
```
aList [i] ← aList [j]
```