CSCI 1470 CS 1
Spring 2026
Suggested Solution to Final Examination

[Q1]

[1]	A	[2]	D				
[3]	D	[4]	D	[5]	D	[6]	B
[7]	D	[8]	A	[9]	A		
[10]	C	[11]	A				

[Q2]

[a]	F	[b]	T	[c]	F	[d]	T	[e]	F
[f]	T	[g]	F	[h]	T	[i]	T	[j]	F

[Q3]

[a]

```
2
456
['1', '4']
987654321
['4', '3', '2']
```

[b]

```
4
[3, 5]
4
[[3, 5], [2, 4]]
4
99
[10, [3, 5], 99]
[10, [3, 5], 99]
[10, [3, 5], [2, 4], 7]
```

[Q4]

[1]

```
integer_count = 0
string_count = 0
```

```
for item in list_1:
    if isinstance(item, int):
        integer_count += 1
    elif isinstance(item, str):
        string_count += 1

print(f"list_1: {list_1}")
print(f"Number of integers in list_1: {integer_count}")
print(f"Number of strings in list_1: {string_count}")
```

[2]

```
list_1 = ["hello world", "It is me", "from Houston"]
print(f"list_1: {list_1}")
count = 0
for s in list_1:
    count += len(s.split())
print(count)

#   or
print(sum(len(s.split()) for s in list_1))
```

[3]

```
dict_2 = {val: key for (key, val) in dict_1.items()}

# Altenatively:
dict_2 = {}
for (key, val) in dict_1.items():
    dict_2[val] = key
print(f"dict_2: {dict_2}")
```

[4]

```
list_2 = [sum(l) for l in list_1]

# Altenatively:
list_2 = []
for l in list_1:
    list_2.append(sum(l))
print(f"list_2: {list_2}")
```

[Q5]

[1]

```
def add_number_lists(list_1, list_2):
    """
```

```
    Returns a list where each element is the sum of the corresponding
    elements in list_1 and list_2.

    If the lists are of different lengths, the resulting list will have the
length of the shorter list.

    Returns:
        A new list containing the sums.
    """
    # The built-in 'zip' function to iterate over corresponding elements
    # from both lists simultaneously. zip stops when the shorter list ends.
    return [x + y for x, y in zip(list_1, list_2)]

def add_number_lists_2(list_1, list_2):
    """
    Alternative version that is less efficient
    """
    length = min(len(list_1), len(list_2))
    result = []
    for i in range(length):
        result.append(list_1[i] + list_2[i])
    return result
```

[2]

```
def merge_and_create_list(list_1, list_2):
    result = []
    for item in list_1:
        if item not in result:
            result.append(item)
    for item in list_2:
        if item not in result:
            result.append(item)
    return result
```