CSE220: Data Structures

1. i)

A Set:

Suppose you are given a multi-dimensional array with dimensions 4x5x4. What is the

linear index of the multidimensional index [2][1][0]?

2\*5\*4+1\*4+0 = 44

B Set:

Suppose you are given a multi-dimensional array with dimensions 5x4x5. What is the

linear index of the multidimensional index [2][1][0]?

2\*4\*5+1\*5+0 = 45

ii)

def find\_sum\_diff(mat):

row,col = mat.shape

result = np.zeros((col-1), dtype = int)

for i in range(col):

sum = 0

for j in range(row):

sum += mat[j][i]

if i > 0:

result[i-1] = sum - prev\_sum

prev\_sum = sum

return result

OR

def find\_sum\_diff\_v2(mat):

row,col = mat.shape

result = np.zeros((col-1), dtype = int)

for i in range(1,col):

sum = 0

for j in range(row):

result[i-1] += mat[j][i]-mat[j][i-1]

return result

def IsSumPossible(head, n):

temp1 = head

while temp1 != None:

temp2 = temp1.next

while temp2 != None:

if temp1.elem + temp2.elem == n:

return True

temp2 = temp2.next

temp1 = temp1.next

return False

1. i) Suppose that an **intermixed** sequence of stack push and pop operations are performed.

The pushes push the integers 0 through 9 in order; the pops print out the return value.

Which of the following sequences could not occur for push and pop operation? Write all

the correct answers.

A. 1 2 3 4 5 6 9 8 7 0 : Correct Answer

B. 0 4 6 5 3 8 1 7 2 9 : Incorrect Answer

C. 1 4 7 9 8 6 5 3 0 2 : Incorrect Answer

D. 2 1 4 3 6 5 8 7 9 0 : Correct answer

**A set :** **B,C [1\*2 = 2]**

**B set : A,D [1\*2 = 2]**

### **Explanation**: [Collapsible]

Option A: Correct

1 2 3 4 5 6 9 8 7 0

Explanation:

| Number | Operation | Print Sequence | Stack |
| --- | --- | --- | --- |
| 0 | 0 pushed into stack |  | 0 |
| 1 | 1 pushed into stack and popped immediately | 1 | 0 |
| 2 | 2 pushed into stack and popped immediately | 2 | 0 |
| 3 | 3 pushed into stack and popped immediately | 3 | 0 |
| 4 | 4 pushed into stack and popped immediately | 4 | 0 |
| 5 | 5 pushed into stack and popped immediately | 5 | 0 |
| 6 | 6 pushed into stack and popped immediately | 6 | 0 |
| 7 | 7 pushed into stack and popped immediately | 7 | 0 |
| 8 | 8 pushed into stack and popped immediately | 8 | 0 |
| 9 | 9 pushed into stack and popped immediately | 9 | 0 |
|  | pop() | 0 | 0 |

Option B: Incorrect

0 4 6 5 3 8 1 7 2 9

| Number | Operation | Print Sequence | Stack |
| --- | --- | --- | --- |
| 0 | 0 pushed into stack and popped immediately | 0 |  |
| 1 | 1 pushed into stack |  | 1 |
| 2 | 2 pushed into stack |  | 2  1 |
| 3 | 3 pushed into stack |  | 3  2  1 |
| 4 | 4 pushed into stack and popped immediately | 4 | 3  2  1 |
| 5 | 5 pushed into stack |  | 5  3  2  1 |
| 6 | 6 pushed into stack and popped immediately | 6 | 5  3  2  1 |
|  | pop() | 5 | 3  2  1 |

Due to the state of the stack, 5 cannot be popped before 3. There we can never get

0 4 6 5 3 8 1 7 2 9 sequence.

Option C: Incorrect

1 4 7 9 8 6 5 3 0 2

| Number | Operation | Print Sequence | Stack |
| --- | --- | --- | --- |
| 0 | 0 pushed into stack |  | 0 |
| 1 | 1 pushed into stack and popped immediately | 1 | 0 |
| 2 | 2 pushed into stack |  | 2  0 |
| 3 | 3 pushed into stack |  | 3  2  0 |
| 4 | 4 pushed into stack and popped immediately | 4 | 3  2  0 |
| 5 | 5 pushed into stack |  | 5  3  2  0 |
| 6 | 6 pushed into stack |  | 6  5  3  2  0 |
| 7 | 7 pushed into stack and popped immediately | 7 | 6  5  3  2  0 |
| 8 | 8 pushed into stack |  | 8  6  5  3  2  0 |
| 9 | 9 pushed into stack and popped immediately | 9 | 8  6  5  3  2  0 |
|  | pop() | 8 | 6  5  3  2  0 |
|  | pop() | 6 | 5  3  2  0 |
|  | pop() | 5 | 3  2  0 |
|  | pop() | 3 | 2  0 |
|  | pop() | 2 | 0 |

Due to the state of the stack, 0 cannot be popped before 2. There we can never get

1 4 7 9 8 6 5 3 0 2 sequence.

Option D: Correct

2 1 4 3 6 5 8 7 9 0

Explanation:

| Number | Operation | Print Sequence | Stack |
| --- | --- | --- | --- |
| 0 | 0 pushed into stack |  | 0 |
| 1 | 1 pushed into stack |  | 1  0 |
| 2 | 2 pushed into stack and popped immediately | 2 | 1  0 |
|  | pop() | 1 | 0 |
| 3 | 3 pushed into stack |  | 3  0 |
| 4 | 4 pushed into stack and popped immediately | 4 | 3  0 |
|  | pop() | 3 | 0 |
| 5 | 5 pushed into stack |  | 5  0 |
| 6 | 6 pushed into stack and popped immediately | 6 | 5  0 |
|  | pop() | 5 | 0 |
| 7 | 7 pushed into stack |  | 7  0 |
| 8 | 8 pushed into stack and popped immediately | 8 | 7  0 |
|  | pop() | 7 | 0 |
| 9 | 9 pushed into stack and popped immediately | 9 | 0 |
|  | pop() | 0 |  |

### ii)

def dance\_pair(st):

female\_q = Queue()

while not st.isEmpty():

a = st.pop()

b = st.peek()

a\_gender, a\_id = id\_gender\_extractor(a)

b\_gender, b\_id = id\_gender\_extractor(b)

if a\_gender != b\_gender and b\_gender != None:

b = st.pop()

print(f'{a\_id} and {b\_id} are paired together')

else:

if a\_gender == 'F':

female\_q.enqueue(a)

else:

if not female\_q.isEmpty():

b = female\_q.dequeue()

b\_gender, b\_id = id\_gender\_extractor(b)

print(f'{b\_id} and {a\_id} are paired together')