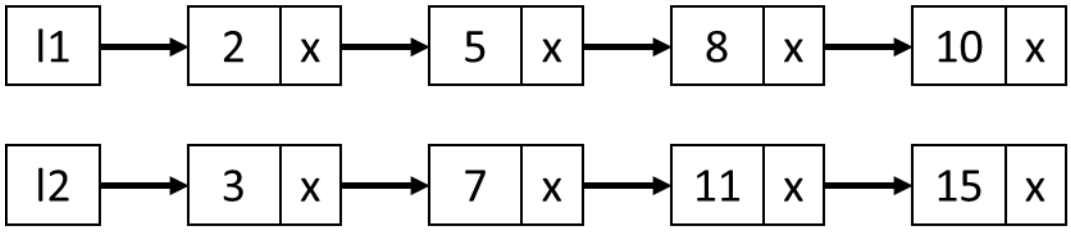
NED UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF COMPUTER AND INFORMATION SYSTEMS ENGINEERING
MIDTERM EXAMINATION SE CS 2023 FALL SEMESTER 2024
CS-218 DATA STRUCTURES AND ALGORITHMS
DATE: 7 OCT 2024 TIME: 1.5 HOURS MARKS: 20

CLO-1: Elaborate fundamental data structures. C2, PLO-1

1. **Calculate** the address of 5 and 4 from the array A= {3,5,1,8,4,2}. Note that each integer is stored in two bytes in the memory and base address of A IS 0FC0H. [2]
2. Consider an algorithm `merge(l1,l2)` that accepts the pointer `l1` to a sorted linked list and the pointer `l2` to another sorted linked list, and returns the head pointer to a linked list which contains ALL values in both the input linked lists in sorted order. **Draw** the linked list returned by the function for the linked lists shown here. [1]



3. **Complete** the code for the function `merge(l1,l2)` to work properly. Rewrite the entire function in your answer book. [6]

```
def merge(l1, l2):
    if l1 is None:
        return ____
    if l2 is None:
        return ____
    a = l1
    b = l2
    if a.data <= b.data:
        h = ListNode(a.data)
        a = a.next
    else:
        h = ListNode(____)
        b = ____
    c = h
    while a is not None ____ b is not None:
        if a.data <= ____:
            c.next = ListNode(____)
            a = ____
        else:
            c.next = ListNode(____)
            b = ____
        c = ____
    if a is None:
        c.next = b
    else:
        c.next = ____
    return h
```

CLO-2: Analyze time and space complexity of algorithms. C4, PLO-2

4. **Figure out** the number of iterations of the while loop for any given value of `n`, and the time complexity of the function. [2]

```
def func1(n):
   c = 0
   x = 0
   while x < n^3:
        c = c + 1
        x = x + 3
   return c
```

5. **Investigate** the given function for its best and worst cases and the relevant time complexities. Assume the pointer `a` points to singly linked list of nodes and is not `None`. [4]

```
def func2(a):
    q = a
    while q is not None:
        if q.data < 0:
            p = q.next
            q.next = None
            return [a, p]
        q = q.next
    return [a, None]
```

CLO-3: Practice with algorithms for widely used computing operations. C3, PLO-3

6. **Modify** the binary search algorithm to find the position of the first occurrence of a value that can occur multiple times in a sorted list. [3]
7. **Develop** the function to calculate the number of live neighbors for Game of Life. Note that the neighbors are the eight cells immediately surrounding a cell: vertically, horizontally, and diagonally. [2]