QUIZ 3 (SECJ2013)
DATA STRUCTURE AND ALGORITHMS
SECTION 02, SEM 1, 2023/2024

PART A (MULTIPLE CHOICE QUESTIONS) **[5 MARKS]**
Answer all the following questions.

1. Referring to Figure 1, which of the code segments below will add a new node, pointed by temp and containing the value 80, at the beginning of the list?

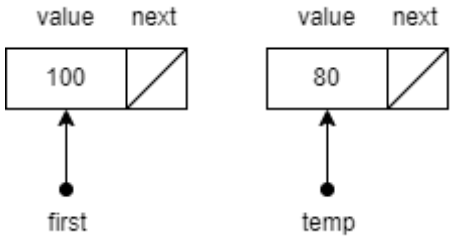


Figure 1

A. Node* temp = new Node; temp->value = 80; temp->next = NULL; first = temp;
B. Node* temp = new Node; temp->value = 80; temp->next = first; first = temp;
C. Node* temp = new Node; first->value = 80; temp->next = first;
D. first = new Node; first->value = 80; first->next = first;isFull()

2. The code segment shown in Figure 2 constructs a linked list with the numbers 22 and 42 as its elements. What should the statement be in the missing section?

```
1   struct NodeType {
2     int data;
3     NodeType *link;
4   };
5
6   NodeType *p, *q;
7
8   p = new NodeType;
9   p->data = 22;
10  q = new NodeType;
11  q->data = 42;
12  ..............  //Statement is missing here
13  q->link = NULL;
```

Figure 2

A. `p = q;`
B. `p->link = new NodeType;`
C. `p->link = q;`
D. `p->link = q->link;`

3. If the elements ‘X’, ‘Y’, ‘11’, and ‘22’ are added to a stack in that order, what would be

the order of removal?

A. '22', '11', 'Y', 'X'
B. 'X', 'Y', '11', '22'
C. '11', '22', 'X', 'Y'
D. 'X', '11', 'Y', '22'

4. Choose the correct final output for the following sequence of stack operations:

push(1)
pop
push(2)
push(3)
pop
pop
push(6)

A. 1 2 3 6
B. 2 3 6
C. 2 3
D. 6

5. Which of the following is the prefix form of **E + H * T** expression?

A. (H*T)+E
B. +EH*T
C. EHT+*
D. +E*HT

PART B (STRUCTURED QUESTIONS) **[35 MARKS]**
Answer all the following questions.

Question 1 **(20 Marks)**

a) Draw the modified linked list based on the initial state of the circular linked list in **Figure 3** after each of the following code segments (i) to (v) has been executed. Code segments are executed in a specific order, so (i) will be executed first, followed by (ii), and so on. Each code segment is dependent on the code segment before it. After each code segment is executed, show the positions of the pointer **`head`**, **`tail`**, and any other pointer(s). Finally, specify the operations carried out by each code segment.

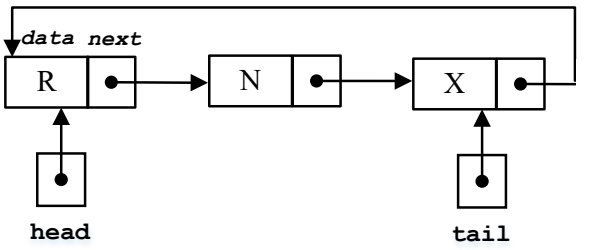
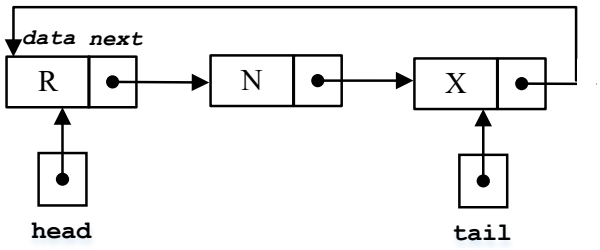


Figure 3: Circular linked list's initial state

Example:

```
1   //Example Code Segment
2   Node *tmp = head;
3
4   while (tmp != tail) {
5     cout << tmp->data;
6     if (tmp != tail)
7       cout << "->";
8     tmp = tmp->next;
9   }
10  cout << tmp->data << endl;
```

Modified linked list: The linked list remains the same as in Figure 3.



The operation performed: Displays all nodes in the list.

(i) (2.5 marks)

```
1   //Code Segment 1
2   Node *n1 = new Node;
3   n1->data = 'T';
4   int nd = 1;
5   tmp = head;
6   while (nd++ != 2 && tmp != tail) {
7     tmp = tmp->next;
8   }
9   tmp->next = n1;
10  n1->next = tail;
```

(ii) (2.5 marks)

```
1   //Code Segment 2
2   tmp = head;
3   while (tmp->next != tail)
4     tmp = tmp->next;
5
6   tail = tmp;
7   tmp = tmp->next;
8   tmp->next = NULL;
9   delete tmp;
10  tail->next = head;
```

(iii) (2.5 marks)

```
1   //Code Segment 3
2   Node *n2 = new Node;
3   n2->data = 'F';
4   n2->next = head;
5   tail->next = n2;
6   tail = n2;
```

(iv) (2.5 marks)

```
1   //Code Segment 4
2   tmp = head;
3   head = head->next;
4   tmp->next = NULL;
5   delete tmp;
6   tail->next = head;
```

b) **Figure 4** depicts five nodes arranged as a doubly linked list, along with four pointer variables (**`tmp`**, **`head`**, **`mid`**, and **`tail`**). Answer all of the questions (i) and (ii). Use only the variables defined in **Figure 4** and do **NOT** define any new variables. Each question is dependent on the one before it.

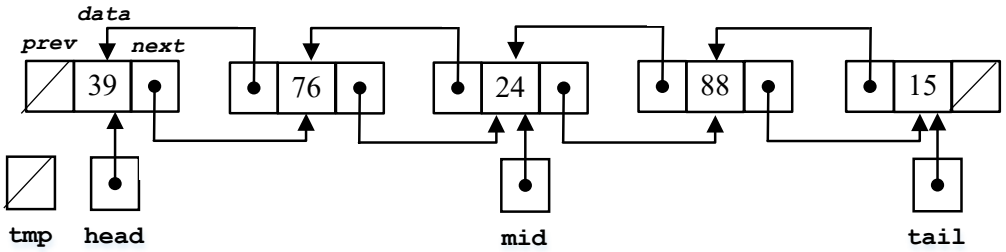


Figure 4: Five-node doubly linked list

(i) Write a C++ code segment to insert a new node, **nn** in **Figure 5**, between the fourth (88) and fifth (15) nodes in **Figure 4**. Assume the pointer variable, **nn** has been defined. **Figure 6** depicts the outcome of this task after the statements in the code segment have been executed. (5 marks)

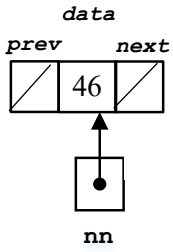


Figure 5: New node to be inserted

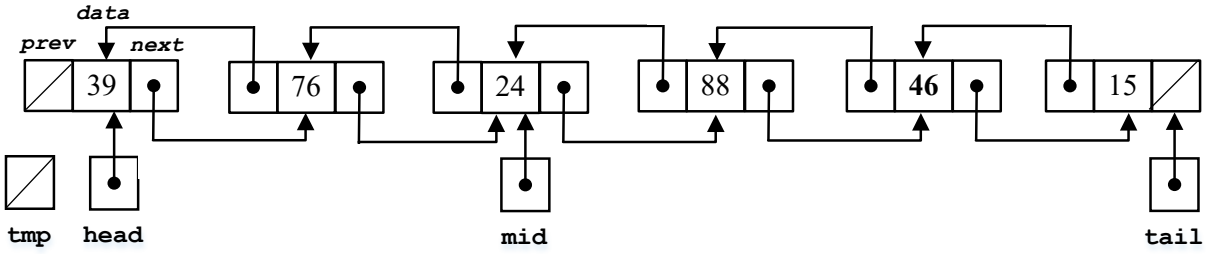


Figure 6: Current nodes in the list after the node is inserted at the fifth position in the list

(ii) Write a C++ code segment to delete the fourth node (88), and connect the third node (24) and fifth node (46) in **Figure 6**. Please release the memory allotted for the fourth node. **Figure 7** depicts the outcome of this task after the statements in the code segment have been executed. (5 marks)

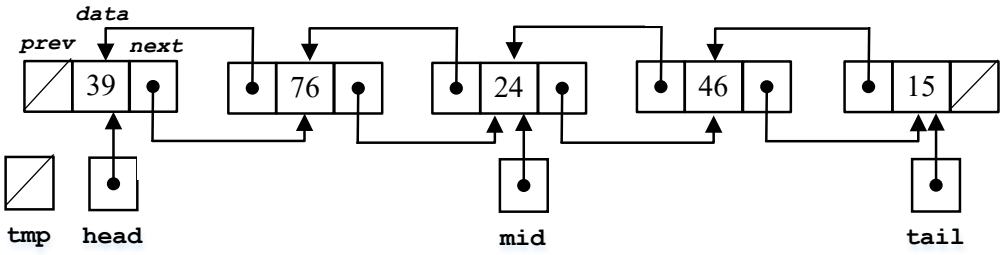


Figure 7: Current nodes in the list after the deletion of the node from the list

Question 2 **(15 Marks)**

Consider the following **Figure 8** is source codes in **Program Segment 1** that will be implemented on stack using array. The size of the stack is 5.

```
1   Stack s;
2   string s1, s2, s3;
3   :
4   :
5   s1 = "Hello";
6   s2 = "Good Day";
7   s3 = "Bye!";
8   s.push(s1);
9   cout<<s.top();
10  s.push(s2+s3);
11  s.push("Arigato");
12  cout<<displayStacks(s);
13  s.pop();
14  s.pop();
15  s.push("Bye Bye!");
16  s.pop();
17  s.push(s3+s2);
```

Figure 8: Program Segment 1 Stack source codes

(i) Show the content of the stack after each implementation of the sources codes based on the situation below. Assume the current content of the stack before statement Line 5 is executed as shown in first stack in the **Table 1** below. (8 marks)

Table 1

[4]								
[3]								
[2]								
[1]								
[0]	Hai							
	Current Content Before Line 5	Content After Line 5-8 are Executed	Content After Line 10	Content After Line 11-12 are Executed	Content After Line 13 is executed	Content After Line 14 is executed	Content After Line 15 – 16 are executed	Content After Line 17 is executed

(ii) Evaluate the following postfix expression using stack concept :

8 64 % 15 * 2 - (7 marks)

Postfix	Ch	Op	Oprn1	Oprn2	Result	Stack
8 64 % 15 * 2 -						
64 % 15 * 2 -						
% 15 * 2 -						
15 * 2 -						
* 2 -						
2 -						
-						