

Q1:{3+2+2=7}
[CLO 3]

- a. Find the best-case and worst-case asymptotic time complexities of the following code segment in terms of n .

```
int count = 0;
if (n % 2) {
    for (int i=1; i <= n; i=i*2)
        for (int j=1; j <= n; j=j+1)
            count++;
} else {
    for (int i=0; i < n; i=i+1)
        for (int j=i+1; j <= n; j=j+1)
            count++;
}
```

- b. What is the exact time-complexity of the given code in terms of n ? Only measure it for the statement `count++`.

```
int count = 0;
for (int i=0; i < n-5; i=i+1)
    for (int j=i+1; j <= n; j=j+1)
        count++;
```

- c. Amina and Bilal are two software developers. They have to choose a sorting algorithm. Amina is of the opinion that the algorithm with less number of statements should be chosen, whereas Bilal says that the algorithm with lesser time complexity should be chosen. What do you think, whose choice is better and why?

$$\{2+2+2=6\}$$

[CLO 2]

- a. What happens when we are writing to a file character by character and the computer shuts down due to load shedding? Do you think there could be a problem with that? If yes, what could be the problem and how can we resolve it?
- b. While opening a file, we separate the file flags (like binary, app, ate, etc.) using the Bitwise OR operator (`|`). How does Bitwise OR work internally? Should there be specific properties for the values of the flags? If yes, what?
- c. What does the `gcount` function do? Give a short example which demonstrates the use of `gcount`.

$\{3+3=6\}$

a. Assume that `itr` is an iterator of a `forward_list<int>` in STL. Which function(s) of the iterator object will be invoked while executing the statement `*itr = 20;`? How can we implement that function(s)?

- b. How does the implementation of pre-increment and post-increment operator differs in forward_list of STL? Explain with the help of an example.

Q4: Assume that you are using array-based implementation of Stack and Queue ADT. How would the arrays of both the ADTs look like after executing the following code? In case of any error in the code, suggest its correction and answer the question based on the corrected code. {3+3=6}

```
stack<int> s(10); queue<int> q(10);
for (int i=1; i<=5; i++) {
    s.push(i*2);
}
s.pop();
for (int i=1; i<=10; i++) {
    q.push(i*3);
}
for (int i=1; i<=3; i++) {
    q.pop();
}
for (int i=1; i<=2; i++) {
    s.push(i*4);
    q.push(i*5);
}
```

//Here, how would the arrays of the stack s and queue q look like?

[CLO 1]

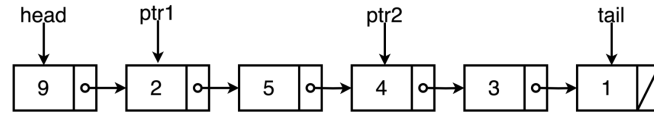
Q5: Convert the following expression into postfix notation: $9 * (2 + 8 - 4 * 3) / 5$
[CLO 2]

{4}

Q6: For this question, write code based on the following linked structures and pointers to the nodes. Write independent code for each part. At the end of each part, the head and tail pointers should always point to the first and last nodes respectively. You may use any number of new pointers, if required.

{1+1+2+2=6}

[CLO 1]



- Display the value 5 without using a loop.
- Display all the values between 5 and 3 using a loop, without comparing the data part.
- Create a new node, store value 4 in it and insert it before the node with value 9.
- Swap the nodes with values 1 and 9 (swap the whole nodes, not just their data parts).

