Q1: How many times the statement count++ will execute in terms of *n* in the following code snippets? {10}
[CLO 3]

a.
```
int count = 0;
for (int i=1; i <= n*2; i=i+1)
  for (int j=1; j <= n; j=j*1)
    count++;
```

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

c.
```
int count = 0;
for (int i=n; i >= 1; i=i/2)
  for (int j=5; j <= n; j=j+1)
    for (int k=1; k <= n/2; k=k+1)
      count++;
```

Q2: {10}
[CLO 2]

a. *Explain the difference between absolute and relative file paths with the help of a simple example.*
b. *Explain the difference between text and binary files. Write code to open a (1) text file (2) binary file.*
c. *How can we move the file pointer of a file to access different parts of a file? Also demonstrate with the help of code.*

Q3: {10}
[CLO 1]

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: {10}

a. *Assume that you are using linked structures-based implementation of Stack, Queue, and List ADT. How would the linked structures 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}*
[CLO 1]

```
stack<int> S;
queue<int> Q;
list<int> L;
for (int i=1; i<=12; i=i++) {
  Q.push(i*2);
  if (i%3 != 0) {
     S.push(i);
     L.push_front(i);
  } else {
     L.push_back(i*2);
  }
}
for (int i=1; i<=3; i++) {
  S.pop();
  Q.pop();
  if (i==2)
     L.pop_front();
  else
     L.pop_back();
}
for (int i=1; i<=2; i++) {
   S.push(i*2);
   Q.push(i*2);
   L.push_front(i*2);
}
```

b. For the list given part a, write client code which uses iterators to display all the even numbers in the list. *{10}*

Q5: Convert the following expression into postfix notation: `2/3*8/2*(4-3+2-1)*3` {10}
[CLO 2]