IDMLI02

– In a Singly Linked List implementation, what do we do when assigning head to null?

Select one:

- Avoid traversing the list.
- Delete all nodes from the list.
- Remove the last node from the list.
- Remove the first node from the list.

IDMLI03

– In an Array-based list, what does this code do to the list?

```
for (int i=pos-1;i<length;i++)
  items[i]=items[i+1];
length--;
```

Select one:

- Remove all item from the list except one.
- Traversing the list.
- Remove an item from the list.
- Duplicate items in the list.

IDMLI04

– In a Singly Linked List implementation, what does this code to to the list?

```
if (!isEmpty())
{
 if (pos == 1)
  head=head.getNext();
 else
 {
  SLNode prevNode=traversing(pos-1);
  SLNode posNode=prevNode.getNext();
  prevNode.setNext(posNode.getNext());
 }
}
```

Select one:

- Search for a node in the list
- Remove the tail node
- Remove the node at the pos position from the list
- Remove the head node

IDMLI05

– In an Array-based list, what does this code do to the list?

```
if (length<maxSize)
{
  length++;
  for (int i=length-1; i>pos-1; i--)
    items[i]=items[i-1];
  items[pos-1]=newItem;
}
```

Select one:

- Insert an item to the list
- Search for an item in the list
- Traversing the list
- Remove an item from the list

IDMLI06

– Consider method F in Java and a singly linked list L below. Suppose that H is the head node of the list L. What is the result if we call F(H)?

Select one:

- 'B'-->'D'-->'F'
- 'A'-->'B'-->'C'-->'D'-->'E'-->'F'
- 'A'-->'C'-->'E'
- 'F'-->'E'-->'D'-->'C'-->'B'-->'A'

IDMLI07

– Consider method F in Java and a singly linked list L below. Suppose that H is the head node of the list L. What is the result if we call F(H)?

Select one:

- ‘A’-->‘B’-->‘C’-->‘D’-->‘E’-->‘F’
- ‘A’-->‘C’-->‘E’
- ‘B’-->‘D’-->‘F’
- ‘F’-->‘E’-->‘D’-->‘C’-->‘B’-->‘A’

IDMLI08

- Consider method F in Java and a singly linked list L below. Suppose that H is the head node of the list L. What is the result if we call F(H)?

Select one:

- ‘E’-->‘C’-->‘A’
- ‘F’-->‘D’-->‘B’
- ‘A’-->‘C’-->‘E’
- ‘B’-->‘D’-->‘F’

IDMLI09

- Consider method F in Java and a singly linked list L below. Suppose that H is the head node of the list L. What is the result if we call F(H)?

Select one:

- ‘B’-->‘D’-->‘F’
- ‘E’-->‘C’-->‘A’
- ‘A’-->‘C’-->‘E’
- ‘F’-->‘D’-->‘B’

IDMLI10

- Consider method F in Java and a singly linked list L below. Suppose that H is the head node of the list L. What is the result if we call F(H.getNext())?

```
public static void F(SLNode node)
{
  if (node!=null)
  {
    System.out.println(node.getData());
    if (node.getNext()!=null)
      F(node.getNext().getNext());
  }
}

L={ 'A' --> 'B' --> 'C' --> 'D' --> 'E' --> 'F' }
H is the head node of L, H= 'A'
```

Select one:

- 'F'-->'D'-->'B'
- 'E'-->'C'-->'A'
- 'B'-->'D'-->'F'
- 'A'-->'C'-->'E'

IDMLI11

– Consider method F in Java and a singly linked list L below. Suppose that H is the head node of the list

L. What is the result if we call F(H.getNext())?

Select one:

- 'B'-->'D'-->'F'
- 'E'-->'C'-->'A'
- 'A'-->'C'-->'E'
- 'F'-->'D'-->'B'

IDMLI12

– What is the number of comparisons needed in the worst case to search for a given node in a Singly Linked List of the length N nodes?

Select one:

- N/2

- log(N)
- N
- Nlog(N)