SET – 1A

Linked Lists

1. In a circular linked list organization, insertion of a record involves modification of

   A. One pointer
   B. Two pointers
   C. Three pointers
   D. No pointer

2. Consider a singly linked list having n nodes. The data items $d_1, d_2, \ldots, d_n$ are stored in the n nodes. Let Y be a pointer to the j^{th} node ($1 \leq j \leq n$) in which d_j is stored. A new data item d stored in a node with address Y is to be inserted. Give an algorithm to insert d into the list to obtain a list having items $d_1, d_2, \ldots\ldots, d_{j-1}, d, d_j, \ldots..d_n$ in that order without using the header.

3. Linked lists are not suitable for data structures for which one of the following problems?

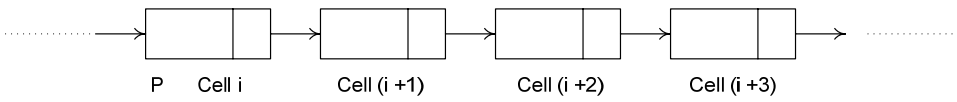
   (A) Insertion sort
   (B) Binary search
   (C) Radix sort
   (D) Polynomial manipulation

4. The concatenation of two lists is to be performed in O(1) time. Which of the following implementations of a list should be used?

   (A) singly linked list (B) doubly linked list

   (C) circular doubly linked list (D) array implementation of list

5. (a) Let p be a pointer as shown in the figure in a singly linked list.



What do the following assignment statements achieve?

q: = p→ next

p → next: = q→ next

q → next: = (q → next) → next

(p → next) →next: = q

Write a constant time algorithm to insert a node with data D just before the node with address p of a singly linked list.

6. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is

   A. $\log_2 n$
   B. $n/2$
   C. $\log_2 n - 1$
   D. n

7. In a circular linked list organization, insertion of a record involves modification of

   A. One pointer
   B. Two pointers
   C. Three pointers
   D. No pointer

8. Consider a singly linked list having n nodes. The data items $d_1, d_2, \ldots, d_n$ are stored in the n nodes. Let Y be a pointer to the j^{th} node ($1 \leq j \leq n$) in which d_j is stored. A new data item d stored in a node with address Y is to be inserted. Give an algorithm to insert d into the list to obtain a list having items $d_1, d_2, \ldots\ldots, d_{j-1}, d, d_j, \ldots..d_n$ in that order without using the header.

9. The following C function takes a singly-linked list of integers as a parameter and rearranges the elements of the list. The function is called with the list containing the integers 1,2,3,4,5,6,7 in the given order. What will be the contents of the list after the function completes execution?

```
struct node  {

    int value;

    struct node *next;

};

void rearrange (struct node *list) {

    struct node *p, *q;

    int temp;



    if (!list || !list -> next) return;

    p = list; q = list -> next;

    while (q)  {

        temp = p -> value; p -> value =q -> value;

        q -> value = temp; p = q -> next;

        q = p ? -> next : 0;

    }

}
```

   (A) 1,2,3,4,5,6,7
   (B) 2,1,4,3,6,5,7
   (C) 1,3,2,5,4,7,6
   (D) 2,3,4,5,6,7,1

10. The data blocks of a very large file in the Unix file system are allocated using

    (A) contiguous allocation
    (B) linked allocation
    (C) indexed allocation
    (D) an extension of indexed allocation

11. The following C function takes a singly linked list of integers as a parameter and rearranges the elements of the list. The list is represented as pointer to structure. The function is called with the list containing integers 1, 2, 3, 4, 5, 6, 7 in the given order. What will be the contents of the list after the function completes?

```
struct node {int value; struct node *next;};

void rearrange(struct node *list) {
        struct node *p, *q;
        int temp;
        if(!list || !list → next) return;
        p = list; q = list → next;
        while(q) {
                temp = p → value;
                p → value = q → value;
                q → value = temp;
                p = q → next;
                q = p? p → next : 0;
        }
}
```

(A) 1, 2, 3, 4, 5, 6, 7
(B) 2, 1, 4, 3, 6, 5, 7
(C) 1, 3, 2, 5, 4, 7, 6
(D) 2, 3, 4, 5, 6, 7, 1

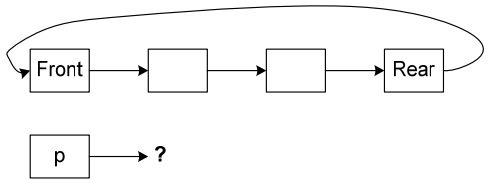
12. Let P be a singly linked list. Let Q be the pointer to an intermediate node x in the list. What is the worst case time complexity of the best-known algorithm to delete the node x from the list?

    (A) O(n)
    (B) $O(\log^2 n)$
    (C) O(log n)
    (D) O(1)

13. Suppose each set is represented as a linked list with elements in arbitrary order. Which of the operations among union, intersection, membership, and cardinality will be the slowest?

(A) Union only
(B) intersection, membership
(C) membership, cardinality
(D) union, intersection

14. Circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations enQueue and deQueue can be performed in constant time?



(A) Rear node
(B) Front node
(C) Not possible with a single pointer
(D) Node next to front