SANFOUNDRY
Global Education & Learning

Data Structure Questions and Answers – Singly Linked List Operations – 3

« Prev Next »

This set of Data Structure Questions and Answers for Experienced people focuses on “Singly Linked Lists Operations – 3”.

1. The following function reverse() is supposed to reverse a singly linked list. There is one line missing at the end of the function.

```
/* Link list node */
struct node
{
    int data;
    struct node* next;
};

/* head_ref is a double pointer which points to head (or start) pointer
  of linked list */
static void reverse(struct node** head_ref)
{
    struct node* prev    = NULL;
    struct node* current = *head_ref;
    struct node* next;
    while (current != NULL)
    {
        next  = current->next;
        current->next = prev;
        prev = current;
        current = next;
    }
    /*ADD A STATEMENT HERE*/
}
```

What should be added in place of “/*ADD A STATEMENT HERE*/”, so that the function correctly reverses a linked list.
a) *head_ref = prev;
b) *head_ref = current;
c) *head_ref = next;

d) *head_ref = NULL;

View Answer

Answer: a

Explanation: *head_ref = prev; At the end of while loop, the prev pointer points to the last node of original linked list.

We need to change *head_ref so that the head pointer now starts pointing to the last node.

2. What is the output of following function for start pointing to first node of following linked list?

```
1->2->3->4->5->6
void fun(struct node* start)
{
    if(start == NULL)
    return;
    printf("%d  ", start->data);
    if(start->next != NULL )
    fun(start->next->next);
    printf("%d  ", start->data);
}
```

a) 1 4 6 6 4 1

b) 1 3 5 1 3 5

c) 1 2 3 5

d) 1 3 5 5 3 1

View Answer

Answer: d

Explanation: fun() prints alternate nodes of the given Linked List, first from head to end, and then from end to head.

If Linked List has even number of nodes, then skips the last node.

3. The following C function takes a simply-linked list as input argument.

It modifies the list by moving the last element to the front of the list and returns

the modified list. Some part of the code is left blank. Choose the correct alternative

to replace the blank line.

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

Node *move_to_front(Node *head)
{
    Node *p, *q;
    if ((head == NULL: || (head->next == NULL))
    return head;
    q = NULL; p = head;
    while (p-> next !=NULL)
    {
        q = p;
        p = p->next;
    }
   _______________________________
  return head;
}
```

a) q = NULL; p->next = head; head = p;
b) q->next = NULL; head = p; p->next = head;
c) head = p; p->next = q; q->next = NULL;
d) q->next = NULL; p->next = head; head = p;

View Answer

Answer: d
Explanation: When while loop completes its execution, node ‘p’ refers to the last node whereas the ‘q’ node refers to the node before ‘p’ in the linked list. q->next=NULL makes q as the last node. p->next=head places p as the first node. the head must be modified to ‘p’ as ‘p’ is the starting node of the list (head=p). Thus the sequence of steps are q->next=NULL, p->next=head, head=p.

4. The following C function takes a single-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?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

View Answer

Answer: b
Explanation: The function rearrange() exchanges data of every node with its next node. It starts exchanging data from the first node itself.

5. 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

View Answer

Answer: d
Explanation: In the worst case, the element to be searched has to be compared with all elements of linked list.

6. Given pointer to a node X in a singly linked list. Only one pointer is given, pointer to head node is not given, can we delete the node X from given linked list?
a) Possible if X is not last node
b) Possible if size of linked list is even
c) Possible if size of linked list is odd
d) Possible if X is not first node

View Answer

Answer: a
Explanation:
Following are simple steps.

```
struct node *temp  = X->next;
X->data  = temp->data;
X->next  = temp->next;
free(temp);
```

7. You are given pointers to first and last nodes of a singly linked list, which of the following operations are dependent on the length of the linked list?
a) Delete the first element
b) Insert a new element as a first element
c) Delete the last element of the list
d) Add a new element at the end of the list
View Answer

Answer: c
Explanation: Deletion of the first element of the list is done in O (1) time by deleting memory and changing the first pointer.
Insertion of an element as a first element can be done in O (1) time. We will create a node that holds data and points to the head of the given linked list. The head pointer was changed to a newly created node.
Deletion of the last element requires a pointer to the previous node of last, which can only be obtained by traversing the list. This requires the length of the linked list.
Adding a new element at the end of the list can be done in O (1) by changing the pointer of the last node to the newly created node and last is changed to a newly created node.

8. In the worst case, the number of comparisons needed to search a singly linked list of length n for a given element is
a) log2 n
b) $^n/_2$
c) log2 n – 1
d) n
View Answer

Answer: d
Explanation: The worst-case happens if the required element is at last or the element is absent in the list. For this, we need to compare every element in the linked list. If n elements are there, n comparisons will happen in the worst case.

Sanfoundry Global Education & Learning Series – Data Structure.

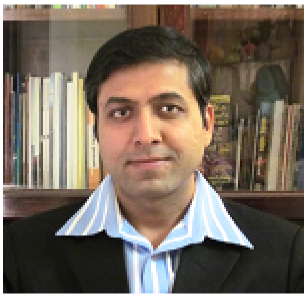
TTo practice all areas of Data Structure for Experienced people, here is complete set of 1000+ Multiple Choice Questions and Answers.

« Prev - Data Structure Questions and Answers – Singly Linked List Operations – 2

» Next - Data Structure Questions and Answers – Singly Linked List

Recommended Posts:

1. Java Algorithms, Problems & Programming Examples
2. Python Programming Examples on Trees
3. C Programming Examples on Arrays
4. Heat Transfer Operations Questions and Answers
5. C Programming Examples on Combinatorial Problems & Algorithms
6. C Programming Examples on Bitwise Operations
7. Data Science Questions and Answers
8. C Programming Examples on Stacks & Queues
9. Data Structures & Algorithms II – Questions and Answers
10. C Tutorials
11. C Programming Examples on Trees
12. C Programming Examples using Recursion
13. C Programming Examples without using Recursion
14. C Programming Examples on Data-Structures
15. Java Programming Examples on Data-Structures
16. Data Structure Questions and Answers
17. C++ Programming Examples on Data-Structures
18. C# Programming Examples on Data Structures
19. Python Programming Examples on Linked Lists
20. C Programming Examples on Linked List



Manish Bhojasia, a technology veteran with 20+ years @ Cisco & Wipro, is Founder and CTO at Sanfoundry. He is Linux Kernel Developer & SAN Architect and is passionate about competency developments in these areas. He lives in Bangalore and delivers focused training sessions to IT professionals in Linux Kernel, Linux Debugging, Linux Device Drivers, Linux Networking, Linux Storage, Advanced C Programming, SAN Storage Technologies, SCSI Internals & Storage Protocols such as iSCSI & Fiber Channel. Stay connected with him @ LinkedIn

Subscribe Sanfoundry Newsletter and Posts

Name*

Email*

Subscribe