**Student Id: (Solution) Section: BAI-3A Marks: 15**

**Quiz#01 - Paper A**

**Question#01**

1. You are given the head of a linked list and an integer k. Return the head of the linked list after swapping the values of the k-th node from the beginning and the k-th node from the end of the list.

**Input:**

* Head: 1 -> 2 -> 3 -> 4 -> 5
* k: 2

**Output:**

* Head: 1 -> 4 -> 3 -> 2 -> 5

|  |
| --- |
| Node\* swapNodes(Node\* head, int k) {  Node\* kBegNode = head;  Node\* kEndNode = head;  Node\* temp = head;    for (int i = 1; i < k; ++i) {  kBegNode = kBegNode->next;  }  temp = kBegNode;  while (kBegNode->next) {  kBegNode = kBegNode->next;  kEndNode = kEndNode->next;  }  int swapValue = temp->val;  temp->val = kEndNode->val;  kEndNode->val = swapValue;  return head;  } |

1. You are given the head of a singly linked list. Write a function to delete all nodes whose values are greater than a specified integer n. After deleting these nodes, the function should return the head of the updated linked list.

### Input:

* Head: 1 -> 7 -> 3 -> 9 -> 2 -> 5 -> 6
* n: 4

### Output:

* Head: 1 -> 3 -> 2

|  |
| --- |
| Node\* deleteNodes (Node\* head, int n) {  // Dummy node -> edge cases  Node\* dummy = new Node(0);  dummy->next = head;    Node\* current = dummy;  // Traverse the list  while (current->next) {  if (current->next->val > n) {  // Delete the node  Node\* temp = current->next;  current->next = current->next->next;  delete temp; // Free the memory  } else {  current = current->next;  }  }  // Return the new head  Node\* newHead = dummy->next;  delete dummy; // Free the dummy node  return newHead;  } |

**Question#02**

Consider the content of an integer array as below:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
| 20 | 3 | 9 | 13 | 11 | 10 | 2 | 1 | 12 | 23 |

How the content of the array will look like after the fourth iteration of the outer loop in the following cases:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Selection Sort | | | | | | | | | |
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
| 1 | 2 | 3 | 9 | 11 | 10 | 13 | 20 | 12 | 23 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bubble Sort | | | | | | | | | |
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
| 3 | 9 | 2 | 1 | 10 | 11 | 12 | 13 | 20 | 23 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Insertion Sort | | | | | | | | | |
| [0] | [1] | [2] | [3] | [4] | [5] | [6] | [7] | [8] | [9] |
| 3 | 9 | 11 | 13 | 20 | 10 | 2 | 1 | 12 | 23 |

**Question#03 [marks**

You are given a singly linked list with the following values:

10 -> 7 -> 5 -> 8 -> 2 -> 6 -> 3 -> 4

Answer the following questions:

1. Identify the sorting algorithm being used to sort this singly linked list.

Insertion sort

1. Write the time complexity of the sorting algorithm.

O(n2)

Dry Run:s

|  |
| --- |
| **Initial List:**  10 -> 7 -> 5 -> 8 -> 2 -> 6 -> 3 -> 4  **List after first iteration:**  7 -> 10 -> 5 -> 8 -> 2 -> 6 -> 3 -> 4  **List after second iteration:**  5 -> 7 -> 10 -> 8 -> 2 -> 6 -> 3 -> 4 |