1.Write a function “insert\_any()” for inserting a node at any given position of the linked list. Assume

position starts at 0.

|  |
| --- |
| #include <iostream> |
|  |  |
|  | using namespace std; |
|  |  |
|  | int main() |
|  | { |
|  | class Node |
|  | { |
|  | public: |
|  | int data; |
|  | Node \* next; |
|  |  |
|  | void insert\_any(Node\* prev\_node,int new\_data) |
|  | { |
|  | if(prev\_node==NULL) |
|  | { |
|  | cout<<"invalid"; |
|  | return; |
|  | } |
|  | Node\* n=new Node(); |
|  | n->data=new\_data; |
|  | n->next=prev\_node->next; |
|  | prev\_node->next=n; |
|  | } |
|  | }; |
|  |  |
|  |  |
|  | } |

2. Write a function “delete\_beg()” for deleting a node from the beginning of the linked list.

|  |
| --- |
| #include <iostream> |
|  |  |
|  | using namespace std; |
|  |  |
|  | class Node { |
|  | int data; |
|  | class Node\* next; |
|  | }; |
|  |  |
|  |  |
|  | Node\* delete\_beg(struct Node\* head) |
|  | { |
|  | if (head == NULL) |
|  | return NULL; |
|  |  |
|  | Node\* temp = head; |
|  | head = head->next; |
|  |  |
|  | delete temp; |
|  |  |
|  | return head; |
|  | } |

3.Write a function “delete\_end()” for deleting a node from the end of the linked list.

|  |
| --- |
| #include <iostream> |
|  |  |
|  | using namespace std; |
|  |  |
|  | class Node { |
|  | int data; |
|  | class Node\* next; |
|  |  |
|  | Node\* delete\_end(struct Node\* head) |
|  | { |
|  | if (head == NULL) |
|  | return NULL; |
|  |  |
|  | Node\* temp = head; |
|  | while(temp->next!=NULL) |
|  | { |
|  | temp=temp->next; |
|  | } |
|  |  |
|  | delete temp; |
|  |  |
|  | return head; |
|  | } |
|  | }; |