Assignment - 3

1. Write a function "insert_any()" for inserting a node at any given position of the linked list. Assume position starts at 0.

```
typedef struct node
     {
        int data;
        Struct node *link;
     };
     Void insert_any();
     Void display();
     Void main()
     {
        int choice;
        int cont = 1;
       header = (struct node * malloc(size of (node));
        clrscr();
       header → data = NULL;
       header → link = NULL;
      while (cont = = 1)
      {
        printf("\n Insert at any position\n");
        printf("\n 2. Display linked list\n");
        Printf("\n Enter your choice:");
        Scanf("%d", &choice);
        Switch (choice) {
            case 1: insert_any();
                     break;
            case 2: display();
                     break;
          }
```

Scanned by CamScanner

```
Printf("\n\n Do you want to continue ? (1/0) :");        5
Scanf("%d", &cont);
  }
  getch();
}
Void insert_any()
{
  int data_value, key;
  printf("\n Enter data of the node :");
  Scanf("%d", & data_Value);
  pf("\n Enter data of the node after which new node is to
                              be inserted:");
  sf("%d", &key);
  temp=(~~struct~~ node *)malloc(sizeof(node));
   ptr = header;
   while(ptr → link!=NULL  && ptr → data!=key )
    {
      ptr = ptr → link;
    }
    if(ptr → data  ==key)
     {
       temp → data = data_Value;
       temp → link = ptr → link;
       ptr → link =temp;
       }
     else
       {
        pf("\n Value %d not found \n",key);
       }
}
void display()
{ pf("\n contents of the linked list are :\n");
    ptr =header;
    while(ptr → link != NULL)
     {
        ptr = ptr → link;
   } }  pf("%d", ptr → date);
```

2. Write a function "delete_beg()" for deleting a node from the beginning of the linked list.

```
typedef struct node
  {
   int data
   struct node *link;
  };
  nodeptr;
  void delete_beg();
  Void main()
   {
     int choice
     header=(node *)malloc(sizeof (node));
     clrscr();
      head->data =NULL;
       header-> link =NULL;
     if (header->link ==NULL)
      {
        pf("\n Empty Linked list. Deletion not possible
      }
      else
      {  ptr=head->link;
          head->link=ptr->link;
          free(ptr);
          pf("\n Node deleted from beg\n");
       }
   }
  void display()
   {pf("\n linked list is:\n");
     ptr=head;
     while(ptr->link!=NULL)
     {ptr =ptr->link;
      pf("%d", ptr->data);
```

Scanned by CamScanner

3. Write a function "delete_end()" for deleting a node from the end of the linked list. ⑦

```
typedef struct node
 {
  int data
  struct node * link;
 };
 node ptr, ptr1;
 void delete_end();
 void main()
 {
 head = (node *)(malloc(sizeof (node));
 head -> data = NULL;
 head -> link = NULL;
 if (head -> link == NULL)
 {
 pf ("\n Empty linked list. Deletion not possible.\n");
 }
   else
    {
      ptr = head;
      while(ptr -> link != NULL)
       {
        ptr1 = ptr;
        ptr = ptr -> link;
       }
      ptr1 -> link = ptr -> link;
      free(ptr);
      pf ("\n Node deleted from the end.\n");
 }    }
```

Scanned by CamScanner