ASSIGNMENT - 3

(1)

```
void insert_any ()
{
    int data_value, key;
    printf (" Enter data of the node: ");
    scanf ("%d", &data_value);
    printf (" Enter data of the node after which
            new node is to be inserted: ");
    scanf ("%d", &key);
    temp = (struct node *) malloc (sizeof (struct 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
void display ()
 {
  printf (" Contents of linked list are: \n");
    ptr = header;
    while (ptr -> link != NULL)
    {
      ptr = ptr -> link;
      printf ("%d", ptr -> data);
    }
}
```

② void delete_beg()
{
struct node *toDelete;
if (head == NULL)
{
printf("List is already empty.");
}
else
{
toDelete = head;
head = head->next;
printf("Data Deleted = %d \n", toDelete->data);
free(toDelete);
printf("First Node deleted \n");
}
}

③ void delete_end()
{
struct node *toDelete, *secondLastNode;
if (head == NULL)
{
printf("List is already empty.");
}
else
{ toDelete = head;
secondLastNode = head;
while(toDelete->next != NULL)
{ secondLastNode = toDelete;
toDelete = toDelete->next;
}
if(toDelete == head)
{ head = NULL; }
else
{ secondLastNode->next = NULL; }