LAB 5a:

```
//Single Linked List insertion
#include <stdio.h>
#include<stdlib.h>


typedef struct Node {
   int data;
   struct Node *next;
}Node;

void InsertAtBeginning( Node **head_ref,int new_data);
void InsertAtEnd( Node **head_ref,int new_data);
void Insert( Node **prev_node,int new_data,int pos);
void PrintList(Node * next);

void InsertAtBeginning( Node **head_ref,int new_data)
{
   struct Node* new_node=(struct Node*)malloc(sizeof( Node));
   new_node->data=new_data;
   new_node->next=*head_ref;
   *head_ref=new_node;
}

void InsertAtEnd(Node **head_ref,int new_data)
{
   struct Node* new_node=(struct Node*)malloc(sizeof( Node));
   Node *last=*head_ref;
   new_node->data=new_data;
   new_node->next=NULL;
   if (*head_ref==NULL)
   {
      *head_ref=new_node;
      return ;
   }
   while (last->next!=NULL)
      last=last->next;
   last->next=new_node;

}

void Insert(Node **head_ref,int new_data,int pos)
{
   if (*head_ref ==NULL)
   {
      printf("Cannot  be NULL\n");
      return;
   }
   Node *temp = *head_ref;
   Node *newNode = ( Node *) malloc (sizeof ( Node));
   newNode->data = new_data;
   newNode->next = NULL;

    while (--pos>0)
          {
           temp = temp->next;
```

```
            }
            newNode->next = temp->next;
    temp->next = newNode;
}


void PrintList(Node *node)
{
   while (node!=NULL)
   {
      printf("%d\n",node->data);
      node=node->next;
   }
}


int main()
{
   int ch,new,pos;
   Node* head=NULL;
   printf("SHREYA S RUDAGI\n");
   printf("1BM22CS267\n\n\n");
   while(ch!=5)
   {
   printf("Menu\n");
   printf("1.Insert at beginning\n");
   printf("2.Insert at a specific position\n");
   printf("3.Insert at end\n");
   printf("4.Display linked list\n");
   printf("5.Exit\n");
   printf("Enter your choice\n");
   scanf("%d",&ch);
   switch(ch)
   {
      case 1:
      {
      printf("Enter the data you want to insert at beginning\n");
      scanf("%d",&new);
      InsertAtBeginning(&head,new);
      break;
      }
      case 2:
      {
      printf("Enter the data and position at which you want to insert \n");
      scanf("%d%d",&new,&pos);
      Insert(&head,new,pos);
      break;
      }
      case 3:
      {
      printf("Enter the data  you want to insert at end\n");
      scanf("%d",&new);
      InsertAtEnd(&head,new);
      break;
      }
      case 4:
      {
```

```
            printf("Created linked list is:\n");
            PrintList(head);
            break;
        }
        case 5:
        {
            return 0;
            break;
        }
        case 6:
        {
            printf("Invalid data!");
            break;
        }
        }
}
return 0;
}
#include <stdio.h>
#include<stdlib.h>


typedef struct Node {
   int data;
   struct Node *next;
}Node;

void InsertAtBeginning( Node **head_ref,int new_data);
void InsertAtEnd( Node **head_ref,int new_data);
void Insert( Node **prev_node,int new_data,int pos);
void PrintList(Node * next);

void InsertAtBeginning( Node **head_ref,int new_data)
{
   struct Node* new_node=(struct Node*)malloc(sizeof( Node));
   new_node->data=new_data;
   new_node->next=*head_ref;
   *head_ref=new_node;
}

void InsertAtEnd(Node **head_ref,int new_data)
{
   struct Node* new_node=(struct Node*)malloc(sizeof( Node));
   Node *last=*head_ref;
   new_node->data=new_data;
   new_node->next=NULL;
   if (*head_ref==NULL)
   {
      *head_ref=new_node;
      return ;
   }
   while (last->next!=NULL)
      last=last->next;
   last->next=new_node;

}
```

```
void Insert(Node **head_ref,int new_data,int pos)
{
   if (*head_ref ==NULL)
   {
      printf("Cannot  be NULL\n");
      return;
   }
   Node *temp = *head_ref;
   Node *newNode = ( Node *) malloc (sizeof ( Node));
   newNode->data = new_data;
   newNode->next = NULL;

     while (--pos>0)
            {
             temp = temp->next;
            }
            newNode->next = temp->next;
    temp->next = newNode;
}


void PrintList(Node *node)
{
   while (node!=NULL)
   {
      printf("%d\n",node->data);
      node=node->next;
   }
}


int main()
{
   int ch,new,pos;
   Node* head=NULL;
   printf("SHREYA S RUDAGI\n");
   printf("1BM22CS267\n\n\n");
   while(ch!=5)
   {
   printf("Menu\n");
   printf("1.Insert at beginning\n");
   printf("2.Insert at a specific position\n");
   printf("3.Insert at end\n");
   printf("4.Display linked list\n");
   printf("5.Exit\n");
   printf("Enter your choice\n");
   scanf("%d",&ch);
   switch(ch)
   {
      case 1:
      {
      printf("Enter the data you want to insert at beginning\n");
      scanf("%d",&new);
      InsertAtBeginning(&head,new);
      break;
      }
      case 2:
```

```
      {
      printf("Enter the data and position at which you want to insert \n");
      scanf("%d%d",&new,&pos);
      Insert(&head,new,pos);
      break;
      }
      case 3:
      {
      printf("Enter the data  you want to insert at end\n");
      scanf("%d",&new);
      InsertAtEnd(&head,new);
      break;
      }
      case 4:
      {
         printf("Created linked list is:\n");
         PrintList(head);
         break;
      }
      case 5:
      {
         return 0;
         break;
      }
      case 6:
      {
         printf("Invalid data!");
         break;
      }
      }
}
return 0;
}
#include <stdio.h>
#include<stdlib.h>


typedef struct Node {
   int data;
   struct Node *next;
}Node;

void InsertAtBeginning( Node **head_ref,int new_data);
void InsertAtEnd( Node **head_ref,int new_data);
void Insert( Node **prev_node,int new_data,int pos);
void PrintList(Node * next);

void InsertAtBeginning( Node **head_ref,int new_data)
{
   struct Node* new_node=(struct Node*)malloc(sizeof( Node));
   new_node->data=new_data;
   new_node->next=*head_ref;
   *head_ref=new_node;
}

void InsertAtEnd(Node **head_ref,int new_data)
{
```

```
   struct Node* new_node=(struct Node*)malloc(sizeof( Node));
   Node *last=*head_ref;
   new_node->data=new_data;
   new_node->next=NULL;
   if (*head_ref==NULL)
   {
      *head_ref=new_node;
      return ;
   }
   while (last->next!=NULL)
      last=last->next;
   last->next=new_node;

}

void Insert(Node **head_ref,int new_data,int pos)
{
   if (*head_ref ==NULL)
   {
      printf("Cannot  be NULL\n");
      return;
   }
   Node *temp = *head_ref;
   Node *newNode = ( Node *) malloc (sizeof ( Node));
   newNode->data = new_data;
   newNode->next = NULL;

     while (--pos>0)
            {
              temp = temp->next;
            }
            newNode->next = temp->next;
    temp->next = newNode;
}


void PrintList(Node *node)
{
   while (node!=NULL)
   {
      printf("%d\n",node->data);
      node=node->next;
   }
}


int main()
{
   int ch,new,pos;
   Node* head=NULL;
   printf("SHREYA S RUDAGI\n");
   printf("1BM22CS267\n\n\n");
   while(ch!=5)
   {
   printf("Menu\n");
   printf("1.Insert at beginning\n");
   printf("2.Insert at a specific position\n");
```

```
   printf("3.Insert at end\n");
   printf("4.Display linked list\n");
   printf("5.Exit\n");
   printf("Enter your choice\n");
   scanf("%d",&ch);
   switch(ch)
   {
      case 1:
      {
      printf("Enter the data you want to insert at beginning\n");
      scanf("%d",&new);
      InsertAtBeginning(&head,new);
      break;
      }
      case 2:
      {
      printf("Enter the data and position at which you want to insert \n");
      scanf("%d%d",&new,&pos);
      Insert(&head,new,pos);
      break;
      }
      case 3:
      {
      printf("Enter the data  you want to insert at end\n");
      scanf("%d",&new);
      InsertAtEnd(&head,new);
      break;
      }
      case 4:
      {
         printf("Created linked list is:\n");
         PrintList(head);
         break;
      }
      case 5:
      {
         return 0;
         break;
      }
      case 6:
      {
         printf("Invalid data!");
         break;
      }
      }
}
return 0;
}
```

```
SHREYA S RUDAGI
1BM22CS267


Menu
1.Insert at beginning
2.Insert at a specific position
3.Insert at end
4.Display linked list
5.Exit
Enter your choice
1
Enter the data you want to insert at beginning
1
Menu
1.Insert at beginning
2.Insert at a specific position
3.Insert at end
4.Display linked list
5.Exit
Enter your choice
3
Enter the data  you want to insert at end
2
Menu
1.Insert at beginning
2.Insert at a specific position
3.Insert at end
4.Display linked list
5.Exit
Enter your choice
2
Enter the data and position at which you want to insert
3
1
Menu
1.Insert at beginning
2.Insert at a specific position
3.Insert at end
4.Display linked list
5.Exit
Enter your choice
4
Created linked list is:
1
3
2
Menu
1.Insert at beginning
2.Insert at a specific position
3.Insert at end
4.Display linked list
5.Exit
Enter your choice
```