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)
{
   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)
{
   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;
    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;
}
```

```
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
3
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
5
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
6
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:
3
5
6
Menu
1.Insert at beginning
2.Insert at a specific position
3.Insert at end
4.Display linked list
5.Exit
Enter your choice
```

Deletion

```
#include <stdio.h>
#include<stdlib.h>


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

void InsertAtBeginning( Node **head_ref,int new_data);
void DeleteAtBeginning( Node **head_ref);
void DeleteAtEnd( Node **head_ref);
void Delete( Node **prev_node,int pos);
void PrintList(Node * next);

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

void DeleteAtBeginning( Node **head_ref)
{
   Node *ptr;
if(head_ref == NULL)
{
printf("\nList is empty");
}
else
{
ptr = *head_ref;
*head_ref = ptr->next;
free(ptr);
printf("\n Node deleted from the beginning ...");

}

}
```

```
void DeleteAtEnd(Node **head_ref)
{
    Node *ptr,*ptr1;

if(*head_ref == NULL)

{

printf("\nlist is empty");

}

else if((*head_ref)-> next == NULL)

{

free(*head_ref);

*head_ref= NULL;

printf("\nOnly node of the list deleted ...");

}

else

{

ptr = *head_ref;

while(ptr->next != NULL)

{

ptr1 = ptr;

ptr = ptr ->next;

}

ptr1->next = NULL;

free(ptr);
```

```
printf("\n Deleted Node from the last ...");

}

}
void Delete(Node **head_ref, int pos)
{
   Node *temp = *head_ref, *prev;

   if (temp == NULL)
   {
      printf("\nList is empty");
      return;
   }

   if (pos == 1)
   {
      *head_ref = temp->next;
      free(temp);
      printf("\nDeleted node with position %d", pos);
      return;
   }

   for (int i = 0; temp != NULL && i < pos - 1; i++)
   {
      prev = temp;
      temp = temp->next;
   }

   if (temp == NULL)
   {
      printf("\nPosition out of range");
      return;
   }

   prev->next = temp->next;
   free(temp);
   printf("\nDeleted node with position %d", pos);
}
void PrintList(Node *node)
{
   while (node!=NULL)
   {
      printf("%d\n",node->data);
```

```
        node=node->next;
    }
}


int main()
{
    int ch,new,pos;
    Node* head=NULL;
    while(ch!=6)
    {
    printf("Menu\n");
    printf("1.Create a linked list\n");
    printf("2.Delete at beginning\n");
    printf("3.Delete at a specific position\n");
    printf("4..Delete at end\n");
    printf("5..Display linked list\n");
    printf("6..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:
        {
        DeleteAtBeginning(&head);
        break;
        }
        case 3:
        {
        printf("Enter the position at which you want to delete \n");
        scanf("%d",&pos);
        Delete(&head,pos);
        break;
        }
        case 4:
        {
        DeleteAtEnd(&head);
```

```
        break;
        }
        case 5:
        {
            printf("Created linked list is:\n");
            PrintList(head);
            break;
        }
        case 6:
        {
            return 0;
            break;
        }
        default:
        {
            printf("Invalid data!");
            break;
        }
        }
}
return 0;
}
```

```
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
1
Enter the data you want to insert at beginning
2
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
1
Enter the data you want to insert at beginning
3
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
1
Enter the data you want to insert at beginning
4
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
1
Enter the data you want to insert at beginning
5
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
1
Enter the data you want to insert at beginning

6
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
```

```
Enter your choice
1
Enter the data you want to insert at beginning

6
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
5
Created linked list is:
6
5
4
3
2
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
4

 Deleted Node from the last ...Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
5
Created linked list is:
6
5
4
3
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
5
Created linked list is:
6
5
4
3
Menu
1.Create a linked list
2.Delete at beginning
```

```
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
3
Enter the position at which you want to delete
2

Deleted node with position 2Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
5
Created linked list is:
6
4
3
Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
2

 Node deleted from the beginning ...Menu
1.Create a linked list
2.Delete at beginning
3.Delete at a specific position
4..Delete at end
5..Display linked list
6..Exit
Enter your choice
5
Created linked list is:
4
3
```