:

#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;

}

Output:

