#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 == 0)

{

\*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;

}

Output:



