AIM: Implement Singly Linked List ADT.

CODE:

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

};

struct node \*head;

void begInsert();

void endInsert();

void begDelete();

void endDelete();

void count();

void display();

void main ()

{

int choice = 0;

clrscr();

while(choice != 7)

{

printf("\n\n---Implementation of Singly Linked List---\n\n");

printf("\n1.Insertion at beginning\n2.Insertion at end\n3.Deletion from beginning\n4.Deletion from end\n5.Number of nodes\n6.Display\n7.Exit\n");

printf("\nEnter your choice: ");

scanf("%d",&choice);

switch(choice)

{

case 1:

begInsert();

break;

case 2:

endInsert();

break;

case 3:

begDelete();

break;

case 4:

endDelete();

break;

case 5:

count();

break;

case 6:

display();

break;

case 7:

exit(0);

break;

default:

printf("Please enter valid choice");

}

}

getch();

}

void begInsert()

{

struct node \*ptr;

int item;

ptr = (struct node \*) malloc(sizeof(struct node ));

if(ptr == NULL)

{

printf("\nOVERFLOW");

}

else

{

printf("\nEnter value: ");

scanf("%d",&item);

ptr->data = item;

ptr->next = head;

head = ptr;

printf("\nNode inserted");

}

}

void endInsert()

{

struct node \*ptr,\*temp;

int item;

ptr = (struct node\*)malloc(sizeof(struct node));

if(ptr == NULL)

{

printf("\nOVERFLOW");

}

else

{

printf("\nEnter value: ");

scanf("%d",&item);

ptr->data = item;

if(head == NULL)

{

ptr -> next = NULL;

head = ptr;

printf("\nNode inserted");

}

else

{

temp = head;

while (temp -> next != NULL)

{

temp = temp -> next;

}

temp->next = ptr;

ptr->next = NULL;

printf("\nNode inserted");

}

}

}

void begDelete()

{

struct node \*ptr;

if(head == NULL)

{

printf("\nList is empty\n");

}

else

{

ptr = head;

head = ptr->next;

free(ptr);

printf("\nNode deleted from the begining ...\n");

}

}

void endDelete()

{

struct node \*ptr,\*ptr1;

if(head == NULL)

{

printf("\nlist is empty");

}

else if(head -> next == NULL)

{

head = NULL;

free(head);

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

}

else

{

ptr = head;

while(ptr->next != NULL)

{

ptr1 = ptr;

ptr = ptr ->next;

}

ptr1->next = NULL;

free(ptr);

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

}

}

void count()

{

struct node \*ptr;

int count = 0;

ptr = head;

if(head == NULL)

{

printf("\nList is empty");

}

while(ptr != NULL)

{

count++;

ptr = ptr -> next;

}

printf("\nThe total number of nodes are: %d", count);

}

void display()

{

struct node \*ptr;

ptr = head;

if(ptr == NULL)

{

printf("Nothing to print");

}

else

{

printf("\nPrinting values . . . . .\n");

while (ptr!=NULL)

{

printf("\t%d",ptr->data);

ptr = ptr -> next;

}

}

}

OUTPUT:  










