|  |
| --- |
|  |
| #include<iostream> |
|  | #include <stdio.h> |
|  | #include <conio.h> |
|  | #include <stdlib.h> |
|  |  |
|  | using namespace std; |
|  |  |
|  | struct node |
|  | { |
|  | int data; |
|  | struct node \*next; |
|  | }; |
|  | struct node \*head; |
|  |  |
|  | void insert\_begin(); |
|  | void deletion\_begin(); |
|  | void deletion\_end(); |
|  | void elementdisplay(); |
|  | int main() |
|  | { |
|  | insert\_end(); |
|  | deletion\_begin(); |
|  | deletion\_end(); |
|  | return 0; |
|  | } |
|  | void insertion\_end()#insert at end of linked list |
|  | { |
|  | int item; |
|  | struct node \*nptr,\*temp; |
|  | nptr= (struct node \*)malloc(sizeof(struct node)); |
|  | cout << "Enter a element:"; |
|  | cin >> item; |
|  | nptr->data = item; |
|  | temp=head; |
|  | if (head == NULL) |
|  | { |
|  | head=nptr; |
|  | head->next = NULL; |
|  | } |
|  | else |
|  | { |
|  | while(temp->next!=NULL) |
|  | { |
|  | temp=temp->next; |
|  | } |
|  | temp->next=nptr; |
|  | nptr->next=NULL; |
|  | } |
|  | } |
|  | void deletion\_begin()#deletion of element at beginning of list |
|  | { |
|  | struct node \*nptr; |
|  | if(head->next==NULL) |
|  | { |
|  | nptr=head; |
|  | head=NULL; |
|  | dealloc(nptr); |
|  | } |
|  | else |
|  | { |
|  | nptr=head; |
|  | head=nptr->next; |
|  | dealloc(nptr); |
|  | } |
|  | } |
|  | void deletion\_end() |
|  | { |
|  | struct node \*nptr,\*nptrtemp; |
|  | if(head->next==NULL) |
|  | { |
|  | nptr=head; |
|  | head=NULL; |
|  | free(nptr); |
|  | } |
|  | else |
|  | { |
|  | nptr=head; |
|  | while(nptr->next!=NULL) |
|  | { |
|  | nptrtemp=nptr; |
|  | nptr=nptr->next; |
|  | } |
|  | nptrtemp->next=NULL; |
|  | dealloc(nptr); |
|  | } |
|  | } |
|  | void elementdisplay(){ |
|  | struct node \*nptr; |
|  | nptr=head; |
|  | if(head == NULL) |
|  | { |
|  | cout << "List is empty"; |
|  | } |
|  | else |
|  | { |
|  | cout << "The values in list are:"; |
|  | while(nptr!=NULL) |
|  | { |
|  | cout << nptr->data; |
|  | nptr=nptr->next; |
|  | } |
|  | } |
|  | } |
|  |  |