|  |
| --- |
| /\*Description: Implementation of Doubly Linked List |
|  | \* Learner: ARSHEE QURESHI |
|  | \* created on: 7th SEPTEMBER 2017 |
|  | \*/ |
|  |  |
|  |  |
|  | #include<stdio.h> |
|  | #include<stdlib.h> |
|  |  |
|  | typedef struct linked\_list{ |
|  | int data; |
|  | struct linked\_list \*next; |
|  | struct linked\_list \*prev; |
|  | }node; |
|  |  |
|  | void print(node \*q) |
|  | { |
|  | node \*ptr; |
|  | ptr=q; |
|  | printf("\n"); |
|  | while(ptr!=NULL) |
|  | { |
|  | printf("%d\t",ptr->data); |
|  | ptr=ptr->next; |
|  | } |
|  | } |
|  | void insbeg(node \*\*q,int no) |
|  | { |
|  | node \*temp,\*ptr; |
|  | temp=\*q; |
|  | ptr=(node\*)malloc(sizeof(node)); |
|  | ptr->data=no; |
|  | ptr->prev=NULL; |
|  | if(temp==NULL) |
|  | ptr->next=NULL; |
|  | else |
|  | { |
|  | ptr->next=temp; |
|  | temp->prev=ptr; |
|  | } |
|  | \*q=ptr; |
|  | printf("\nELEMENTS OF LINKLIST AFTER INSERTION\n"); |
|  | print(\*q); |
|  | } |
|  |  |
|  | void insend(node \*\*q,int no) |
|  | { |
|  | node \*ptr,\*temp; |
|  | ptr=(node\*)malloc(sizeof(node)); |
|  | ptr->data=no; |
|  | ptr->next=NULL; |
|  | temp=\*q; |
|  | if(temp==NULL){ |
|  | \*q=ptr; |
|  | ptr->prev=NULL; |
|  | } |
|  | else |
|  | { |
|  | while(temp->next!=NULL) |
|  | temp=temp->next; |
|  | temp->next=ptr; |
|  | ptr->prev=temp; |
|  | } |
|  | printf("\nELEMENTS OF LINKLIST AFTER INSERTION\n"); |
|  | print(\*q); |
|  | } |
|  | void insafter(node \*q,int no) |
|  | { |
|  | int loc,k; |
|  | node \*temp,\*ptr; |
|  | temp=q; |
|  | ptr=(node\*)malloc(sizeof(node\*)); |
|  | ptr->data=no; |
|  | printf("ENTER LOCATION WHERE THE NO.IS TO BE INSERTED: "); |
|  | scanf("%d",&loc); |
|  | if(loc==1){ |
|  | printf("\nPlease use the insert at beginning option\n"); |
|  | return ; |
|  | } |
|  | for(k=1;k<loc;k++) |
|  | { |
|  | if(temp==NULL) |
|  | printf("\nELEMENTS ARE LESS THAN PROVIDED LOCATION\n"); |
|  | else |
|  | { |
|  | //old=temp; |
|  | temp=temp->next; |
|  | } |
|  | } |
|  | if(temp==NULL) |
|  | { |
|  | printf("\nPlease use the insert at end option\n"); |
|  | return ; |
|  | } |
|  | temp->prev->next=ptr; |
|  | ptr->prev=temp->prev; |
|  | ptr->next=temp; |
|  | temp->prev=ptr; |
|  | printf("\nELEMENTS OF LINKLIST AFTER INSERTION\n"); |
|  | print(q); |
|  | } |
|  | void del(node \*\*q,int no) |
|  | { |
|  | int f=0; |
|  | node \*temp; |
|  | temp=\*q; |
|  |  |
|  | while(temp!=NULL) |
|  | { |
|  | if(temp->data==no) |
|  | { |
|  | f=1; |
|  | if(temp==\*q) |
|  | { |
|  | \*q=temp->next; |
|  | if(temp->next!=NULL) |
|  |  |
|  | temp->next->prev=NULL; |
|  |  |
|  | } |
|  | else |
|  | { |
|  | temp->prev->next=temp->next; |
|  | if(temp->next!=NULL) |
|  | temp->next->prev=temp->prev; |
|  | } |
|  | free(temp); |
|  | break; |
|  | } |
|  | else |
|  | { |
|  | temp=temp->next; |
|  | } |
|  | } |
|  |  |
|  | if(f==0) |
|  | printf("\nTHE GIVEN NUMBER IS NOT FOUND\n"); |
|  |  |
|  | printf("\nELEMENTS OF LINKLIST AFTER DELETION\n"); |
|  | print(\*q); |
|  | } |
|  | void traverse(node \*q) |
|  | { |
|  | printf("\nTRAVERSING LINKLIST\n\n"); |
|  | print(q); |
|  | printf("\n\nEND OF LINKLIST\n"); |
|  | } |
|  |  |
|  |  |
|  |  |
|  |  |
|  | int main() |
|  | { |
|  | node \*start,\*ptr,\*temp; |
|  | int i,n,j,no,c; |
|  | printf("ENTER NUMBER OF NODES: "); |
|  | scanf("%d",&n); |
|  | if(n<1) |
|  | { |
|  | printf("\n Invalid number iserted\n"); |
|  | return 0; |
|  | } |
|  | printf("\nENTER NODE NUMBER 1: "); |
|  | start=(node\*)malloc(sizeof(node)); |
|  | scanf("%d",&start->data); |
|  | start->prev=NULL; |
|  | temp=start; |
|  | for(i=1;i<n;i++) |
|  | { |
|  | ptr=(node\*)malloc(sizeof(node)); |
|  | printf("\nENTER NODE NUMBER %d: ",i+1); |
|  | scanf("%d",&ptr->data); |
|  | temp->next=ptr; |
|  | ptr->prev=temp; |
|  | temp=ptr; |
|  | } |
|  | temp->next=NULL; |
|  | do{ |
|  | printf("\nENTER YOUR CHOICE\n"); |
|  | printf("\n1-INSERTION\n2-DELETION\n3-TRAVERSE\n4-Exit\n\n"); |
|  | scanf("%d",&j); |
|  | switch(j) |
|  | { |
|  | case 1: |
|  | printf("\nENTER THE NUMBER TO BE INSERTED: "); |
|  | scanf("%d",&no); |
|  | printf("\nENTER 1 TO INSERT AT THE BEGINING\n"); |
|  | printf("ENTER 2 TO INSERT AT THE END\n"); |
|  | printf("ENTER 3 TO INSERT AT A SPECIFIED LOCATION\n"); |
|  | scanf("%d",&c); |
|  | switch(c) |
|  | { |
|  | case 1: |
|  | insbeg(&start,no); |
|  | break; |
|  | case 2: |
|  | insend(&start,no); |
|  | break; |
|  | case 3: |
|  | insafter(start,no); |
|  | break; |
|  | default: |
|  | printf("\nInvalid Choice."); |
|  | break; |
|  | } |
|  | break; |
|  | case 2: |
|  | printf("\nENTER THE NUMBER TO BE DELETED: "); |
|  | scanf("%d",&no); |
|  | del(&start,no); |
|  | break; |
|  | case 3: |
|  | traverse(start); |
|  | break; |
|  | case 4: |
|  | exit(0); |
|  | default: |
|  | printf("\nInvalid Input."); |
|  | break; |
|  | } |
|  | }while(1); |
|  | return 0; |
|  | } |
|  |  |
|  | /\*ENTER NUMBER OF NODES: 4 |
|  |  |
|  | ENTER NODE NUMBER 1: 1 |
|  |  |
|  | ENTER NODE NUMBER 2: 2 |
|  |  |
|  | ENTER NODE NUMBER 3: 3 |
|  |  |
|  | ENTER NODE NUMBER 4: 4 |
|  |  |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1-INSERTION |
|  | 2-DELETION |
|  | 3-TRAVERSE |
|  | 4-Exit |
|  |  |
|  | 1 |
|  |  |
|  | ENTER THE NUMBER TO BE INSERTED: 15 |
|  |  |
|  | ENTER 1 TO INSERT AT THE BEGINING |
|  | ENTER 2 TO INSERT AT THE END |
|  | ENTER 3 TO INSERT AT A SPECIFIED LOCATION |
|  | 1 |
|  |  |
|  | ELEMENTS OF LINKLIST AFTER INSERTION |
|  |  |
|  | 15 1 2 3 4 |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1-INSERTION |
|  | 2-DELETION |
|  | 3-TRAVERSE |
|  | 4-Exit |
|  |  |
|  | 1 |
|  |  |
|  | ENTER THE NUMBER TO BE INSERTED: 25 |
|  |  |
|  | ENTER 1 TO INSERT AT THE BEGINING |
|  | ENTER 2 TO INSERT AT THE END |
|  | ENTER 3 TO INSERT AT A SPECIFIED LOCATION |
|  | 2 |
|  |  |
|  | ELEMENTS OF LINKLIST AFTER INSERTION |
|  |  |
|  | 15 1 2 3 4 25 |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1-INSERTION |
|  | 2-DELETION |
|  | 3-TRAVERSE |
|  | 4-Exit |
|  |  |
|  | 1 |
|  |  |
|  | ENTER THE NUMBER TO BE INSERTED: 35 |
|  |  |
|  | ENTER 1 TO INSERT AT THE BEGINING |
|  | ENTER 2 TO INSERT AT THE END |
|  | ENTER 3 TO INSERT AT A SPECIFIED LOCATION |
|  | 3 |
|  | ENTER LOCATION WHERE THE NO.IS TO BE INSERTED: 5 |
|  |  |
|  | ELEMENTS OF LINKLIST AFTER INSERTION |
|  |  |
|  | 15 1 2 3 35 4 25 |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1-INSERTION |
|  | 2-DELETION |
|  | 3-TRAVERSE |
|  | 4-Exit |
|  |  |
|  | 1 |
|  |  |
|  | ENTER THE NUMBER TO BE INSERTED: 45 |
|  |  |
|  | ENTER 1 TO INSERT AT THE BEGINING |
|  | ENTER 2 TO INSERT AT THE END |
|  | ENTER 3 TO INSERT AT A SPECIFIED LOCATION |
|  | 3 |
|  | ENTER LOCATION WHERE THE NO.IS TO BE INSERTED: 1 |
|  |  |
|  | Please use the insert at beginning option |
|  |  |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1-INSERTION |
|  | 2-DELETION |
|  | 3-TRAVERSE |
|  | 4-Exit |
|  |  |
|  | 1 |
|  |  |
|  | ENTER THE NUMBER TO BE INSERTED: 55 |
|  |  |
|  | ENTER 1 TO INSERT AT THE BEGINING |
|  | ENTER 2 TO INSERT AT THE END |
|  | ENTER 3 TO INSERT AT A SPECIFIED LOCATION |
|  | 3 |
|  | ENTER LOCATION WHERE THE NO.IS TO BE INSERTED: 8 |
|  |  |
|  | Please use the insert at end option |
|  |  |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1-INSERTION |
|  | 2-DELETION |
|  | 3-TRAVERSE |
|  | 4-Exit |
|  |  |
|  | 3 |
|  |  |
|  | TRAVERSING LINKLIST |
|  |  |
|  |  |
|  | 15 1 2 3 35 4 25 |
|  |  |
|  | END OF LINKLIST |
|  |  |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1-INSERTION |
|  | 2-DELETION |
|  | 3-TRAVERSE |
|  | 4-Exit |
|  |  |
|  | 2 |
|  |  |
|  | ENTER THE NUMBER TO BE DELETED: 2 |
|  |  |
|  | ELEMENTS OF LINKLIST AFTER DELETION |
|  |  |
|  | 15 1 3 35 4 25 |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1-INSERTION |
|  | 2-DELETION |
|  | 3-TRAVERSE |
|  | 4-Exit |
|  |  |
|  | 2 |
|  |  |
|  | ENTER THE NUMBER TO BE DELETED: 66 |
|  |  |
|  | THE GIVEN NUMBER IS NOT FOUND |
|  |  |
|  | ELEMENTS OF LINKLIST AFTER DELETION |
|  |  |
|  | 15 1 3 35 4 25 |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1-INSERTION |
|  | 2-DELETION |
|  | 3-TRAVERSE |
|  | 4-Exit |
|  |  |
|  | 6 |
|  |  |
|  | Invalid Input. |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1-INSERTION |
|  | 2-DELETION |
|  | 3-TRAVERSE |
|  | 4-Exit |
|  |  |
|  | 4 |
|  |  |
|  |  |
|  | ------------------ |
|  | (program exited with code: 0) |
|  | Press return to continue |
|  |  |
|  | \*/ |