|  |
| --- |
| /\*Description: Implementation of Singly Linked List |
|  | \* Learner: ARSHEE QURESHI |
|  | \* created on: 31st AUGUST 2017 |
|  | \*/ |
|  |  |
|  | #include<stdio.h> |
|  | #include<stdlib.h> |
|  |  |
|  | typedef struct linked\_list{ |
|  | int data; |
|  | struct linked\_list \*next; |
|  | }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; |
|  | if(temp==NULL) |
|  | ptr->next=NULL; |
|  | else |
|  | ptr->next=temp; |
|  | \*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; |
|  | else |
|  | { |
|  | while(temp->next!=NULL) |
|  | temp=temp->next; |
|  | temp->next=ptr; |
|  | } |
|  | printf("\nELEMENTS OF LINKLIST AFTER INSERTION\n"); |
|  | print(\*q); |
|  | } |
|  |  |
|  | void insafter(node \*q,int no) |
|  | { |
|  | int loc,k; |
|  | node \*temp,\*ptr,\*old; |
|  | temp=q; |
|  | ptr=(node\*)malloc(sizeof(node)); |
|  | ptr->data=no; |
|  | printf("ENTER LOCATION WHERE THE NO IS TO BE INSERTED: "); |
|  | scanf("%d",&loc); |
|  | for(k=1;k<loc;k++) |
|  | { |
|  | if(temp==NULL) |
|  | printf("\nELEMENTS ARE LESS THAN PROVIDED LOCATION\n"); |
|  | else |
|  | { |
|  | old=temp; |
|  | temp=temp->next; |
|  | } |
|  | } |
|  | ptr->next=temp; |
|  | old->next=ptr; |
|  | printf("\nELEMENTS OF LINKELIST AFTER INSERTION \n"); |
|  | print(q); |
|  | } |
|  |  |
|  | void del(node \*\*q,int no) |
|  | { |
|  | int f=0; |
|  | node \*old,\*temp; |
|  | temp=\*q; |
|  |  |
|  | while(temp!=NULL) |
|  | { |
|  | if(temp->data==no) |
|  | { |
|  | f=1; |
|  | if(temp==\*q) |
|  | \*q=temp->next; |
|  | else |
|  | old->next=temp->next; |
|  | free(temp); |
|  | break; |
|  | } |
|  | else |
|  | { |
|  | old=temp; |
|  | 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"); |
|  | print(q); |
|  | printf("\nIN END OF LINKLIST \n"); |
|  | } |
|  |  |
|  |  |
|  | int main() |
|  | { |
|  | node \*start,\*ptr,\*temp; |
|  | int i,n,j,no,c; |
|  | printf("ENTER NUMBER OF NODES:"); |
|  | scanf("%d",&n); |
|  | printf("\nENTER NODE NUMBER1:"); |
|  | start=(node\*)malloc(sizeof(node)); |
|  | scanf("%d",&start->data); |
|  | 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; |
|  | temp=ptr; |
|  | } |
|  | temp->next=NULL; |
|  | do |
|  | { |
|  | printf("\nENTER YOUR CHOICE\n"); |
|  | printf("\n1.INSERTION\n2.DELETION\n3.TRAVERSE\n4.EXIT\n"); |
|  | scanf("%d",&j); |
|  | switch(j) |
|  | { |
|  | case 1: |
|  | printf("ENTER THE NUMBER TO BE INSERTED"); |
|  | scanf("%d",&no); |
|  | printf("\nENTER 1 TO INSERT AT THE BEGINNING\n"); |
|  | printf("ENTER 2 TO INSERT AT THE END\n"); |
|  | printf("\nENTER 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("invalid 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:3 |
|  |  |
|  | ENTER NODE NUMBER1:10 |
|  |  |
|  | ENTER NODE NUMBER0: 20 |
|  |  |
|  | ENTER NODE NUMBER1: 30 |
|  |  |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1.INSERTION |
|  | 2.DELETION |
|  | 3.TRAVERSE |
|  | 4.EXIT |
|  | 1 |
|  | ENTER THE NUMBER TO BE INSERTED11 |
|  |  |
|  | ENTER 1 TO INSERT AT THE BEGINNING |
|  | ENTER 2 TO INSERT AT THE END |
|  |  |
|  | ENTER 3 TO INSERT AT A SPECIFIED LOCATION |
|  | 1 |
|  |  |
|  | ELEMENTS OF LINKLIST AFTER INSERTION |
|  |  |
|  | 11 10 20 30 |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1.INSERTION |
|  | 2.DELETION |
|  | 3.TRAVERSE |
|  | 4.EXIT |
|  | 1 |
|  | ENTER THE NUMBER TO BE INSERTED22 |
|  |  |
|  | ENTER 1 TO INSERT AT THE BEGINNING |
|  | ENTER 2 TO INSERT AT THE END |
|  |  |
|  | ENTER 3 TO INSERT AT A SPECIFIED LOCATION |
|  | 2 |
|  |  |
|  | ELEMENTS OF LINKLIST AFTER INSERTION |
|  |  |
|  | 11 10 20 30 22 |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1.INSERTION |
|  | 2.DELETION |
|  | 3.TRAVERSE |
|  | 4.EXIT |
|  | 1 |
|  | ENTER THE NUMBER TO BE INSERTED33 |
|  |  |
|  | ENTER 1 TO INSERT AT THE BEGINNING |
|  | 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: 3 |
|  |  |
|  | ELEMENTS OF LINKELIST AFTER INSERTION |
|  |  |
|  | 11 10 33 20 30 22 |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1.INSERTION |
|  | 2.DELETION |
|  | 3.TRAVERSE |
|  | 4.EXIT |
|  | 3 |
|  |  |
|  | TRAVERSING LINKLIST |
|  |  |
|  | 11 10 33 20 30 22 |
|  | IN END OF LINKLIST |
|  |  |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1.INSERTION |
|  | 2.DELETION |
|  | 3.TRAVERSE |
|  | 4.EXIT |
|  | 2 |
|  |  |
|  | ENTER THE NUMBER TO BE DELETED33 |
|  |  |
|  | ELEMENTS OF LINKLIST AFTER DELETION |
|  |  |
|  | 11 10 20 30 22 |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1.INSERTION |
|  | 2.DELETION |
|  | 3.TRAVERSE |
|  | 4.EXIT |
|  | 2 |
|  |  |
|  | ENTER THE NUMBER TO BE DELETED66 |
|  |  |
|  | THE GIVEN NUMBER IS NOT FOUND |
|  |  |
|  | ELEMENTS OF LINKLIST AFTER DELETION |
|  |  |
|  | 11 10 20 30 22 |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1.INSERTION |
|  | 2.DELETION |
|  | 3.TRAVERSE |
|  | 4.EXIT |
|  | 8 |
|  |  |
|  | INVALID INPUT |
|  | ENTER YOUR CHOICE |
|  |  |
|  | 1.INSERTION |
|  | 2.DELETION |
|  | 3.TRAVERSE |
|  | 4.EXIT |
|  | 4 |
|  |  |
|  |  |
|  | ------------------ |
|  | (program exited with code: 0) |
|  | Press return to continue |
|  |  |
|  |  |