**singly-linked-list.c**

//implement singly linked list with the basic function of it.

#include <stdio.h>

#include <stdlib.h>

typedef struct linkedlist {

int data;

struct linkedlist \*next;

}lkdlist;

int isEmpty(lkdlist \*head){

if(!head){

return 1;

}

return 0;

}

int len(lkdlist \*head){

int count = 0;

while(head != NULL){

head = head->next;

count++;

}

return count;

}

void push\_at(lkdlist \*\*head, int new\_data, int index){

if(len(\*head) < index || index < 0){

printf("Index Out of Range %d\n",len(\*head));

return;

}

lkdlist \*new\_node = (lkdlist\*)malloc(sizeof(lkdlist));

new\_node->data = new\_data;

if(index == 0){

new\_node->next = \*head;

\*head = new\_node;

return;

}

int i = 0;

lkdlist \*temp = \*head, \*prev;

do{

prev = temp;

temp = temp->next;

i++;

}while(i < index);

prev->next = new\_node;

new\_node->next = temp;

}

void push\_top(lkdlist \*\*head, int new\_data) {

push\_at(head,new\_data,0);

}

void push\_bottom(lkdlist \*\*head, int new\_data){

int l = len(\*head);

push\_at(head,new\_data,l);

}

int pop\_at(lkdlist \*\*head, int index){

if(isEmpty(\*head)){

printf("No Element Found.\n");

return -1;

}

if(len(\*head) <= index || index < 0){

printf("Index Out of Range.\n");

return -1;

}

lkdlist \*temp = \*head, \*prev;

int i = 0;

int data;

if(i == index){

data = temp->data;

\*head = temp->next;

free(temp);

return data;

}

while(i < index){

prev = temp;

temp = temp->next;

i++;

}

prev->next = temp->next;

data = temp->data;

free(temp);

return data;

}

int pop\_top(lkdlist \*\*head) {

pop\_at(head,0);

}

int pop\_bottom(lkdlist \*\*head){

int l = len(\*head);

pop\_at(head,l-1);

}

void display(lkdlist \*head) {

while (head != NULL) {

printf("%d -> ", head->data);

head = head->next;

}

printf("NULL\n");

}

int scanint(){

int n;

printf("Enter the Number: ");

scanf("%d",&n);

return n;

}

void main(){

int op,data,index;

lkdlist \*ll1 = NULL;

while(1){

printf("|1. Push at Beginning, |2. Push at Index, |3. Push at End,\n");

printf("|4. Pop at Beginning, |5. Pop at Index, |6. Pop at End,\n");

printf("|7. Length, |8. Display List, |9. End the Program,\n");

printf("|->Enter your choice: ");

scanf("%d",&op);

switch(op){

case 1:

push\_top(&ll1,scanint());

break;

case 2:

printf("Enter the Index: ");

scanf("%d",&index);

push\_at(&ll1,scanint(),index);

break;

case 3:

push\_bottom(&ll1,scanint());

break;

case 4:

if(ll1){

printf("Removed %d.\n",pop\_top(&ll1));

}else{

printf("Empty List.\n");

}

break;

case 5:

printf("Enter the Index: ");

scanf("%d",&index);

if(ll1){

printf("Removed %d.\n",pop\_at(&ll1,index));

}else{

printf("Empty List.\n");

}

break;

case 6:

if(ll1){

printf("Removed %d.\n",pop\_bottom(&ll1));

}else{

printf("Empty List.\n");

}

break;

case 7:

printf("The length of the list is %d.\n",len(ll1));

break;

case 8:

display(ll1);

break;

case 9:

printf("Oops... Program Terminated.");

return;

default:

printf("Invalid Input.\n");

}

}

}

**OUTPUT**

PS S:\WorkSpace\CollegeWork\DataStructure\Temp> gcc .\singly-linked-list.c

PS S:\WorkSpace\CollegeWork\DataStructure\Temp> ./a

|1. Push at Beginning, |2. Push at Index, |3. Push at End,

|4. Pop at Beginning, |5. Pop at Index, |6. Pop at End,

|7. Length, |8. Display List, |9. End the Program,

|->Enter your choice: 1

Enter the Number: 12

|1. Push at Beginning, |2. Push at Index, |3. Push at End,

|4. Pop at Beginning, |5. Pop at Index, |6. Pop at End,

|7. Length, |8. Display List, |9. End the Program,

|->Enter your choice: 3

Enter the Number: 16

|1. Push at Beginning, |2. Push at Index, |3. Push at End,

|4. Pop at Beginning, |5. Pop at Index, |6. Pop at End,

|7. Length, |8. Display List, |9. End the Program,

|->Enter your choice: 2

Enter the Index: 1

Enter the Number: 14

|1. Push at Beginning, |2. Push at Index, |3. Push at End,

|4. Pop at Beginning, |5. Pop at Index, |6. Pop at End,

|7. Length, |8. Display List, |9. End the Program,

|->Enter your choice: 8

12 -> 14 -> 16 -> NULL

|1. Push at Beginning, |2. Push at Index, |3. Push at End,

|4. Pop at Beginning, |5. Pop at Index, |6. Pop at End,

|7. Length, |8. Display List, |9. End the Program,

|->Enter your choice: 7

The length of the list is 3.

|1. Push at Beginning, |2. Push at Index, |3. Push at End,

|4. Pop at Beginning, |5. Pop at Index, |6. Pop at End,

|7. Length, |8. Display List, |9. End the Program,

|->Enter your choice: 5

Enter the Index: 1

Removed 14.

|1. Push at Beginning, |2. Push at Index, |3. Push at End,

|4. Pop at Beginning, |5. Pop at Index, |6. Pop at End,

|7. Length, |8. Display List, |9. End the Program,

|->Enter your choice: 6

Removed 16.

|1. Push at Beginning, |2. Push at Index, |3. Push at End,

|4. Pop at Beginning, |5. Pop at Index, |6. Pop at End,

|7. Length, |8. Display List, |9. End the Program,

|->Enter your choice: 4

Removed 12.

|1. Push at Beginning, |2. Push at Index, |3. Push at End,

|4. Pop at Beginning, |5. Pop at Index, |6. Pop at End,

|7. Length, |8. Display List, |9. End the Program,

|->Enter your choice: 8

NULL

|1. Push at Beginning, |2. Push at Index, |3. Push at End,

|4. Pop at Beginning, |5. Pop at Index, |6. Pop at End,

|7. Length, |8. Display List, |9. End the Program,

|->Enter your choice: 9

Oops... Program Terminated.

PS S:\WorkSpace\CollegeWork\DataStructure\Temp>