stack-using-linked-list.c

```
//implement stack data structure using linked list.
#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){
      head = head->next;
      count++;
   }
   return count;
}

int item_at(lkdlist *head,int index){
   if(index >= len(head)){
      printf("index out of range.\n");
      return -1;
   }
   int count = 1;
   while(count <= index){
      head = head->next;
      count++;
   }
   return head->data;
}
int top(lkdlist *head){
   item_at(head,0);
}
int bottom(lkdlist *head){
   item_at(head,len(head)-1);
}

void push(lkdlist **head, int new_data) {
   struct node *new_node = (struct node *)malloc(sizeof(struct node));
   new_node->data = new_data;
   new_node->next = *head;
   *head = new_node;
}
Int pop(lkdlist **head){
   if(!*head){
      printf("No Item left to Remove.");
      return -1;
   }
   lkdlist *temp = *head;
   int data = temp->data;
   *head = temp->next;
   free(temp);
   return data;
}
```

```
void display(lkdlist *head) {
   while (head != NULL) {
      printf("%d -> ", head->data);
      head = head->next;
   }
   printf("NULL\n");
}

void main(){
   lkdlist *stack1 = NULL;
   int op,x;
   while(1){
      printf("1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit.\nEnter Your choice: ");
      scanf("%d",&op);
      switch(op){
         case 1:
            printf("Enter the item: ");
            scanf("%d",&x);
            push(&stack1,x);
            break;
         case 2:
            if(isEmpty(stack1)){
               printf("Underflow.\n");
            }else{
               printf("Removed %d from stack\n",pop(&stack1));
            }
            break;
         case 3:
            if(isEmpty(stack1)){
               printf("Stack is empty.\n");
            }else{
               printf("The elements of stack are: ");
               display(stack1);
            }
            break;
         case 4:
            printf("%d\n",len(stack1));
            break;
         case 5:
            if(isEmpty(stack1)){
               printf("Stack is Empty\n");
            }else{
               printf("Stack is Not Empty\n");
            }
            break;
         case 6:
            printf("Don't worry stack is never going to be full, you have enough space remain.\n");
            break;
         case 7:
            if(isEmpty(stack1)){
               printf("Stack is Empty\n");
            }else{
               printf("%d\n",top(stack1));
            }
            break;
         case 8:
            if(isEmpty(stack1)){
               printf("Stack is Empty\n");
            }else{
               printf("%d\n",bottom(stack1));
            }
            break;
         case 9:
```

```
            exit(0);
        default:
            printf("Invalid Input.\n");
    }
  }
}
```

OUTPUT

```
PS S:\WorkSpace\CollegeWork\DataStructure> gcc .\stack-using-linked-list.c
PS S:\WorkSpace\CollegeWork\DataStructure> ./a
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit. Enter Your choice: 1 12 1 13 1 14 1 15 1
16 1 18 1 109
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit. Enter Your choice: 3
The elements of stack are: 109 -> 18 -> 16 -> 15 -> 14 -> 13 -> 12 -> NULL
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit. Enter Your choice: 2 2 2
Removed 109 from stack
Removed 18 from stack
Removed 16 from stack
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit. Enter Your choice: 3
The elements of stack are: 15 -> 14 -> 13 -> 12 -> NULL
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit. Enter Your choice: 4
4
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit. Enter Your choice: 7
15
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit. Enter Your choice: 8
12
1.Push, 2.Pop, 3.Display, 4. Length, 5. isEmpty, 6. isFull, 7. Top, 8, bottom, 9.Exit. Enter Your choice: 9
PS S:\WorkSpace\CollegeWork\DataStructure>
```