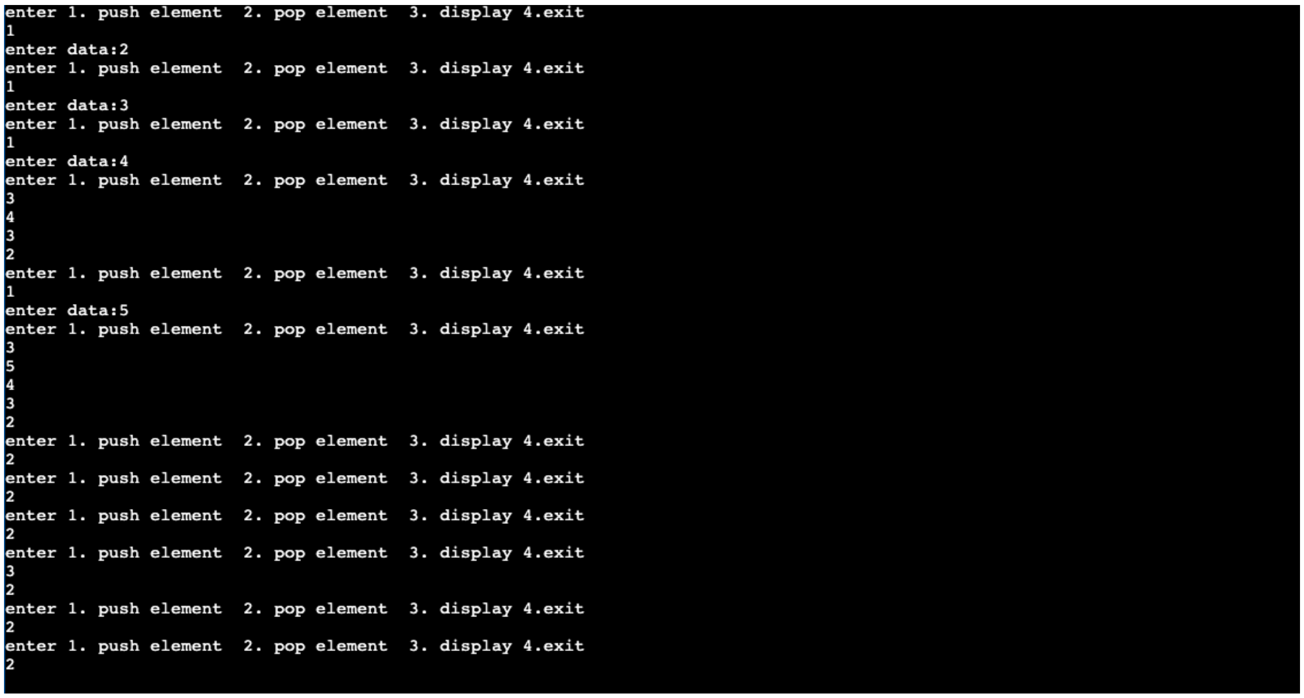
WAP to Implement Single Link List to simulate Stack & Queue Operations

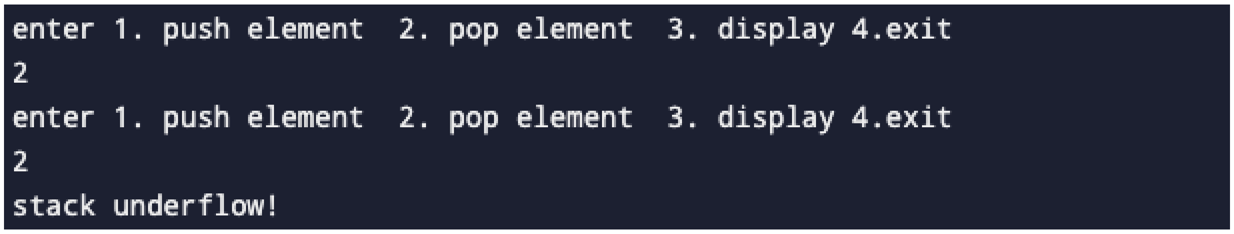
STACK:

```
#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    struct node *next;
};
struct node *head, *temp, *newnode, *p;

void push(){
   newnode=(struct node *)malloc(sizeof(struct node));
   printf("enter data:");
   scanf("%d",&newnode->data);
   if(head==NULL){
        head=temp=newnode;
   }
   else{
        newnode->next=temp;
        head=newnode;
        temp=newnode;
   }
}
void pop(){
    if(head==NULL){
        printf("stack underflow!\n");
    }
    else{
    p=head;
    head=head->next;
    p->next=0;
    free(p);
    }
}
void display(){
    temp=head;
    while(temp!=NULL){
        printf("%d\n",temp->data);
        temp=temp->next;
    }
    temp=head=newnode;
}

int main(){
    head=NULL;
    int c;
    while(1){
    printf("enter 1. push element  2. pop element  3. display 4.exit\n");
    scanf("%d",&c);

    switch(c){
        case 1: push();
                break;
        case 2: pop();
                break;
        case 3: display();
                break;
        case 4: exit(1);
    }
    }
}
```

OUTPUT:



```
enter 1. push element  2. pop element  3. display 4.exit
1
enter data:2
enter 1. push element  2. pop element  3. display 4.exit
1
enter data:3
enter 1. push element  2. pop element  3. display 4.exit
1
enter data:4
enter 1. push element  2. pop element  3. display 4.exit
3
4
3
2
enter 1. push element  2. pop element  3. display 4.exit
1
enter data:5
enter 1. push element  2. pop element  3. display 4.exit
3
5
4
3
2
enter 1. push element  2. pop element  3. display 4.exit
2
enter 1. push element  2. pop element  3. display 4.exit
2
enter 1. push element  2. pop element  3. display 4.exit
2
enter 1. push element  2. pop element  3. display 4.exit
3
2
enter 1. push element  2. pop element  3. display 4.exit
2
enter 1. push element  2. pop element  3. display 4.exit
2
```

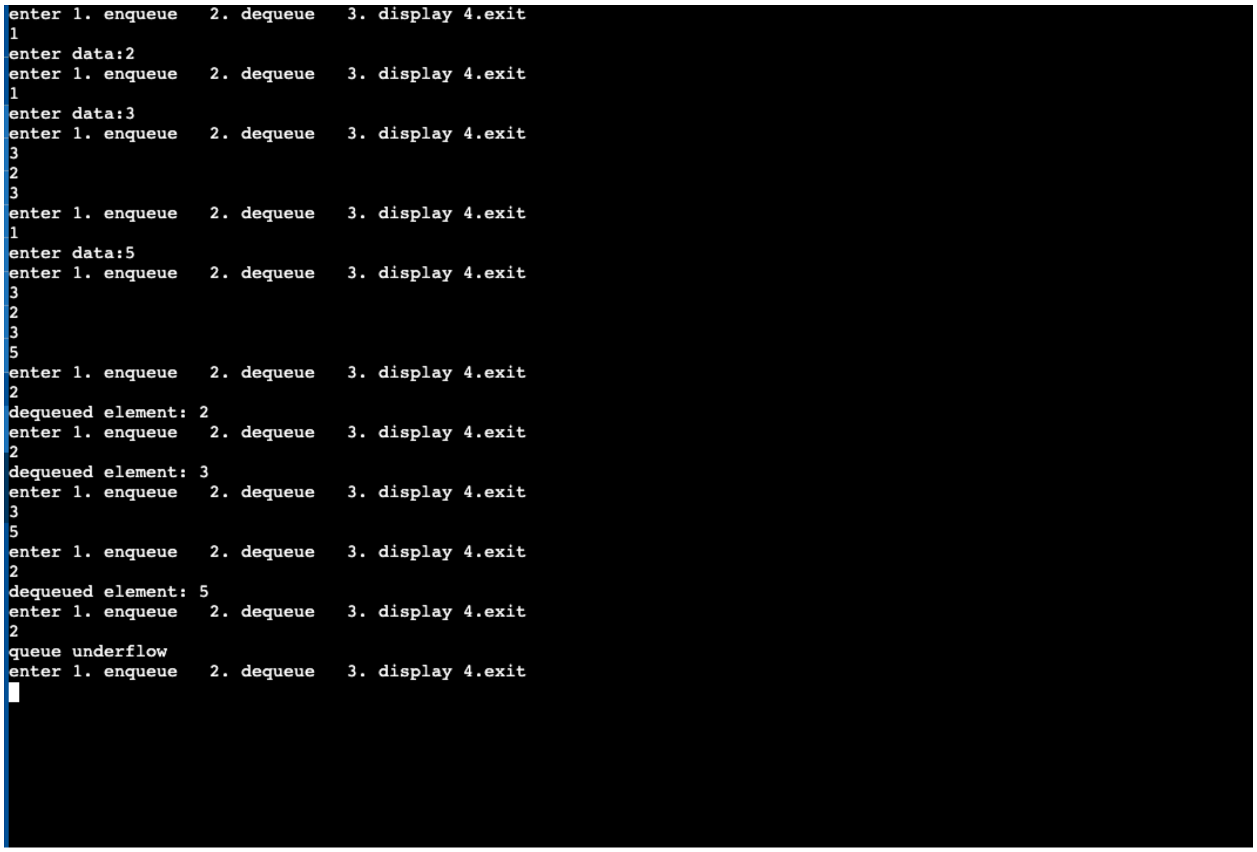


```
enter 1. push element  2. pop element  3. display 4.exit
2
enter 1. push element  2. pop element  3. display 4.exit
2
stack underflow!
```

QUEUE:

```
#include <stdio.h>
#include <stdlib.h>

struct node{
    int data;
    struct node *next;
};
struct node *front, *rear, *newnode, *temp, *p;

void enqueue(){
   newnode=(struct node *)malloc(sizeof(struct node));
   printf("enter data:");
   scanf("%d",&newnode->data);
   if(front==NULL && rear==NULL){
      front=rear=newnode;
   }
   else{
      rear->next=newnode;                          //O(1)
      rear=rear->next; //rear=newnode;
   }
}
void dequeue(){ //delete from beginning
    if(front==NULL){
        printf("queue underflow\n");
    }
    else{
    printf("dequeued element: %d\n",front->data);
    p=front;
    front=front->next;
    p->next=NULL;
    free(p);
    }
}
void display(){
    temp=front;            //temp pointer to traverse and display
    if(rear==0 && front==0){
        printf("Queue is empty\n");
    }
    else{
        while(temp!=NULL){
            printf("%d\n",temp->data);
            temp=temp->next;
        }
    }
}

int main(){
    front=NULL;
    rear=NULL;  //tail
    int c;
    while(1){
    printf("enter 1. enqueue   2. dequeue   3. display 4.exit\n");
    scanf("%d",&c);

    switch(c){
        case 1: enqueue();
                break;
        case 2: dequeue();
                break;
        case 3: display();
                break;
        case 4: exit(1);
    }
    }
}
```

OUTPUT:



```
enter 1. enqueue   2. dequeue   3. display 4.exit
1
enter data:2
enter 1. enqueue   2. dequeue   3. display 4.exit
1
enter data:3
enter 1. enqueue   2. dequeue   3. display 4.exit
3
2
3
enter 1. enqueue   2. dequeue   3. display 4.exit
1
enter data:5
enter 1. enqueue   2. dequeue   3. display 4.exit
3
2
3
5
enter 1. enqueue   2. dequeue   3. display 4.exit
2
dequeued element: 2
enter 1. enqueue   2. dequeue   3. display 4.exit
2
dequeued element: 3
enter 1. enqueue   2. dequeue   3. display 4.exit
3
5
enter 1. enqueue   2. dequeue   3. display 4.exit
2
dequeued element: 5
enter 1. enqueue   2. dequeue   3. display 4.exit
2
queue underflow
enter 1. enqueue   2. dequeue   3. display 4.exit
```

Leetcode Question:

https://leetcode.com/problems/reverse-linked-list/submissions/1160271929/

```c
/**
* Definition for singly-linked list.
* struct ListNode {
*     int val;
*     struct ListNode *next;
* };
*/
struct ListNode* reverseList(struct ListNode* head) {
   struct ListNode* prev = NULL;
   struct ListNode* temp;
   struct ListNode* n;
   temp=head;

   while (temp != NULL) {
       n = temp->next;
       temp->next = prev;
       prev = temp;
       temp = n;
   }

   return prev;
}
```

Sanjana Shetty- 1BM22CS238