Problem Solving
Name: Hari Krishna Shah
VIT ID: 21BCS0167
LINK:
https://drive.google.com/drive/folders/1kMNJngmCzHtqiju-pWVkSBKQDSGXj-qS?usp=sharing

```c
#include<stdio.h>
#include <malloc.h>
//Coded by Hari Krishna Shah

struct Node
{
   int data;
   struct Node *next;
};

void push(int);
void pop();
void display();
void peak();
struct Node *top = NULL;

// Creating a temporary stack to save the stack in reverse order
and to display in reverse order.
//Using the same struct Node for a new stack

struct Node *reverse_top = NULL;
void push_reverse(int);
void reverse();

int main()
{
   int choice = 0, value;
```

```
   printf("\nThis is stack implementation using linked
list.\n");
   while(choice != 5){
      printf("\n\t\t****** This program is made by Hari Krishna
Shah ******\n");
      printf("Welcome to the main menu. \n \
       Enter 1 for push operation\n \
       Enter 2 for pop operation\n \
       Enter 3 to display the stack in reverse order\n \
       Enter 4 for Peak Operation\n \
       Enter 5 to exit\n");
       printf("Enter your choice here: ");
      scanf("%d",&choice);
      switch(choice){
      case 1:{
          printf("Enter the value to be insert: ");
           scanf("%d", &value);
           push(value);
          break;
      }
      case 2:{
          pop();
          break;
      }
      case 3:{
          reverse();
          display();
          break;
      }
      case 4:{
          peak();
          break;
      }
      case 5:{
          printf("Thank you for using the program. This code is
made by Hari Krishna Shah with love.\n");
          break;
      }
      default: printf("\nPlease enter a valid option and try
```

```
again.\n");
      }
      printf("\n");
   }
   return 0;
}
void push(int value)
{
   struct Node *newNode;
   newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = value;
   if(top == NULL){
     newNode->next = NULL;
   }
   else{
     newNode->next = top;
   }
   top = newNode;
   printf("\nInsertion is Successful!!!\n");
}
void pop()
{
   if(top == NULL)
      printf("\nStack is Empty!!!\n");
   else{
      struct Node *temp = top;
      printf("\nDeleted element: %d", temp->data);
      top = temp->next;
      free(temp);
   }
}

void peak(){
     if(top == NULL){
          printf("Stack is empty.\n");
     }
     else{
          printf("The top value is %d.\n", top->data);
     }
```

```
}


void display()
{
   if(reverse_top == NULL){
      printf("\nStack is Empty!!!\n");
   }
   else{
      struct Node *temp = reverse_top;
      while(temp->next != NULL){
           printf("%d--->",temp->data);
           temp = temp ->next;
      }
      printf("%d",temp->data);
   }
}
void push_reverse(int value){
     struct Node *newNode;
   newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = value;
   if(reverse_top == NULL){
     newNode->next = NULL;
   }
   else{
     newNode->next = reverse_top;
   }
   reverse_top = newNode;
}
void reverse(){
     if(top == NULL){
          reverse_top = NULL;
     }
     else{
          struct Node *temp = top;
          while(temp->next != NULL){
               push_reverse(temp->data);
               temp = temp->next;
          }
```

```
            push_reverse(temp->data);
        }

}
```

Note: The program is working fine for all operation. Please be considerate while rewarding marks.

Check the output screen below: