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

struct node
{
 int data;
 struct node *next;
};
//*****************************
struct node* createnode()
{
 int data;
 struct node *newnode =NULL;
 newnode = (struct node *)malloc(sizeof(struct node));
 if(newnode == NULL)
 {
 printf("Memory not allocated ==>> overflow!!!!\n");
 }
 else
 {
  printf("Enter Data ==>> ");
  scanf("%d",&data);
  newnode->data = data;
  newnode->next = NULL;
 }
 return newnode;
}
//*************************************************
 int IS_underflow(struct node*top)
 {
  if(top == NULL)
  {
   printf("STACK UNDERFLOW!!!!!!\n");
   return 0;
  }
  else
  {
  printf("stack is not underflow you can insert data\n");
   return 1;
  }
}
//****************************
int IS_overflow()
{
  struct node* newnode = NULL;
  newnode = (struct node *)malloc(sizeof(struct node));
  if(newnode == NULL)
  {
    printf("STACK OVERFLOW!!!!\n");
    return 0;
  }
  else
  {
  printf("There is space inside stack\n");
  return 1;
  }
}
//*************************************
void Push(struct node**top)
{
 struct node * newnode = *top;

 if(IS_overflow())
 {
  if(newnode == NULL)
  {
   newnode = createnode();
   *top = newnode;
  }

  else
  {
   newnode = createnode();
   newnode->next = *top;
   *top = newnode;
  }
 }
}


//*****************************************
 void Display(struct node * top)
 {
 int check = IS_underflow(top);
 if(check ==1)
 {
  while(top!=NULL)
  {
   printf("%d\n",top->data);
   top = top->next;
  }
 }
}
//***********************************
void Pop(struct node **top)

{
 int check = IS_underflow(*top);
 if(check == 1)
 {
  struct node *newnode = NULL;
  newnode = *top;
  *top = (*top)->next;
  printf("%d is popped out\n",newnode->data);
  free(newnode);
}
}
//**********************************************
void main()
{
  int choice;
  struct node*top =NULL;
  do
  {

printf("1.PUSH\n2.POP\n3.DISPLAY\n4.ISUNDERFLOW\n5.ISOVERFOLW\n0.EXIT\n*****************\nPlese
Enter your choice\n");
   scanf("%d",&choice);
   switch(choice)
   {
    case 1: Push(&top);
            break;
    case 2: Pop(&top);
            break;
    case 3: Display(top);
            break;
    case 4: IS_underflow(top);
           break;
     case 5:IS_overflow();
            break;
   }
   }while(choice != 0);
```

```
}
```