Program 1

Write a program to simulate the working of stack using an array with the following:

a) Push

b) Pop

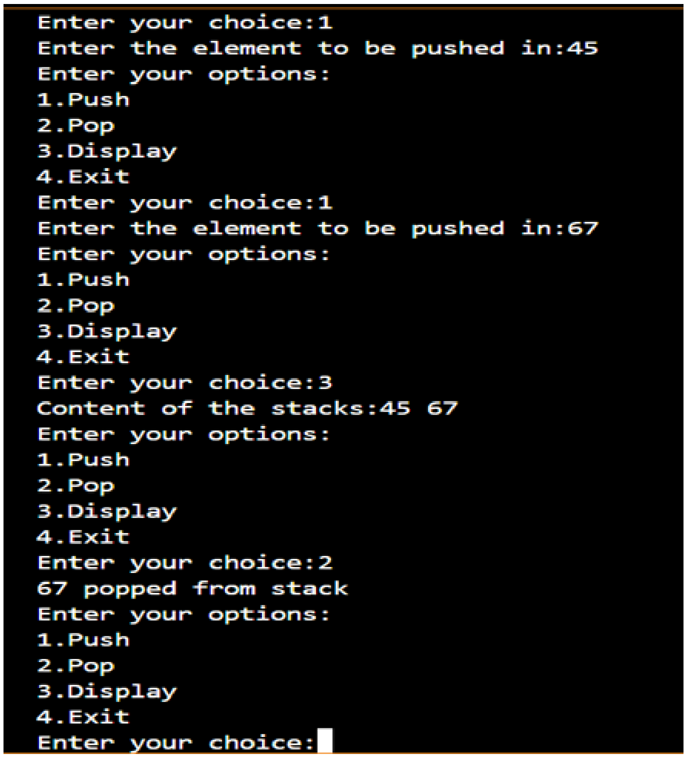
c) Display

The program should print appropriate messages for stack overflow, stack underflow

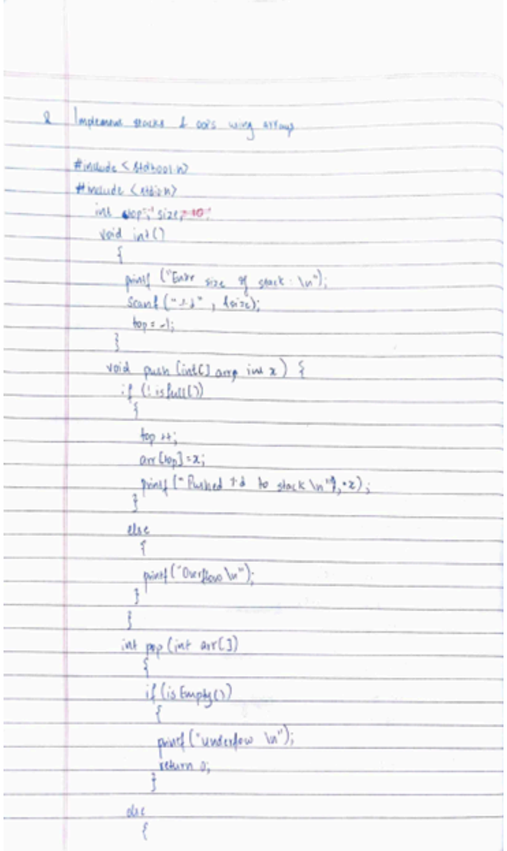
Code:

```
 #include<stdio.h>
#include<stdlib.h>
#define size 5

int top=-1;
int stack[size];
int item;

void push(){
    if(top==size-1){
        printf("Stack Overload\n");
    }
    else{
        top+=1;
        stack[top]=item;
    }
}

int pop(){
    if(top==-1){
        printf("Stack Underflow\n");
    }
    else{
        return stack[top--];
    }
}

void display(){
    if(top==-1){
        printf("Stack is empty!");
    }
    else{
        printf("Content of the stacks:");
        for(int i=0;i<=top;i++){
            printf("%d ",stack[i]);
        }printf("\n");
    }
}
```

```
void main(){
    int choice;
    while(1){
        printf("Enter your options:\n");
        printf("1.Push\n2.Pop\n3.Display\n4.Exit\n");
        printf("Enter your choice:");
        scanf("%d",&choice);
        switch(choice){
            case 1:printf("Enter the element to be pushed in:");scanf("%d",&item);push();break;
            case 2:if(top==-1){
                printf("stack is empty!\n");
            }else{
                printf("%d popped from stack\n", stack[top]);
            }
            pop();
            break;
            case 3:display();
            break;
            case 4:exit(0);
        }
    }
}
```

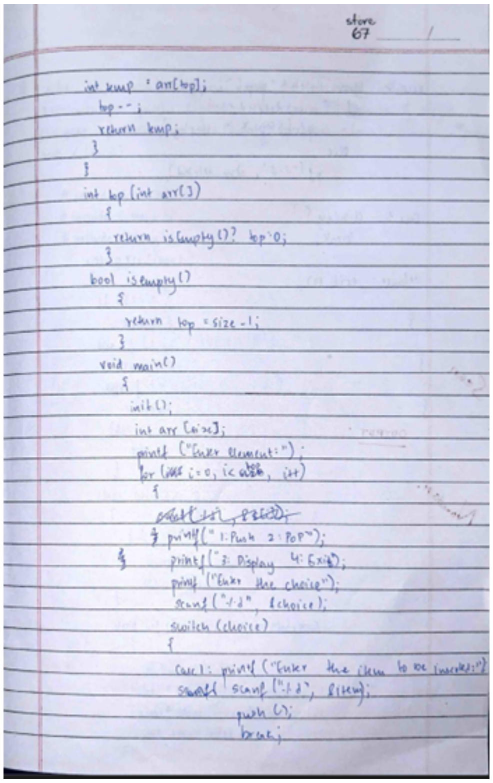


```
Enter your choice:1
Enter the element to be pushed in:45
Enter your options:
1.Push
2.Pop
3.Display
4.Exit
Enter your choice:1
Enter the element to be pushed in:67
Enter your options:
1.Push
2.Pop
3.Display
4.Exit
Enter your choice:3
Content of the stacks:45 67
Enter your options:
1.Push
2.Pop
3.Display
4.Exit
Enter your choice:2
67 popped from stack
Enter your options:
1.Push
2.Pop
3.Display
4.Exit
Enter your choice:
```



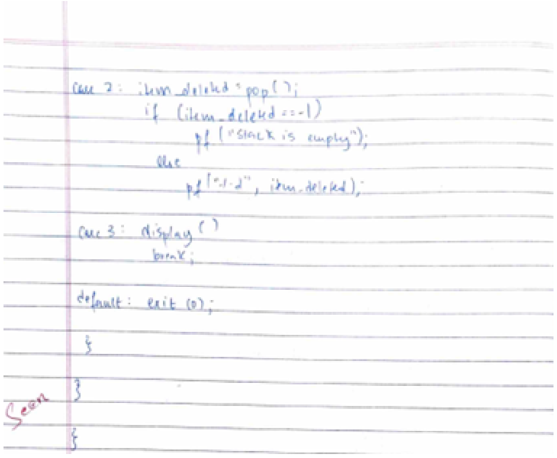
2 Implement stacks & ops using arrays

```
#include <stdbool.h>
#include <stdio.h>
  int ~~stop~~;' size;~~=10~~
  void int()
    {
      printf ("Enter size of stack: \n");
      Scanf ("%d", &size);
      top = -1;
    }
   void push (int[] arr, int x) {
    if (!isfull())
     {
      top ++;
      arr [top] = x;
      printf ("Pushed %d to stack \n", x);
     }
    else
     {
      printf ("Overflow \n");
     }
    }
    int pop (int arr[])
     {
      if (isEmpty())
       {
        printf ("underflow \n");
        return 0;
       }
    else
     {
```



store
67

```
int temp = arr[top];
  top --;
  return temp;
 }
}
int top (int arr[])
   {
    return isEmpty()? top:0;
   }
bool isempty()
   {
     return top = size -1;
   }
 void main()
   {
    init();
    int arr [size];
    printf ("Enter element:");
    for (~~int~~ i=0, i<~~size~~top, i++)
     {
      ~~scanf("%d", &arr[i]);~~
     } printf(" 1:Push  2:PoP");
   }     printf(" 3:Display  4:Exit");
         printf ("Enter the choice");
         scanf ("%d", &choice);
         switch (choice)
         {
          case 1: printf ("Enter the item to be inserted:")
          ~~scanf~~ scanf ("%d", &item);
                   push ();
                   break;
```



```
case 2:  item_deleted = pop();
         if (item_deleted ==-1)
              pf ("stack is empty");
         else
              pf ("%d", item_deleted);

case 3:  display ()
         break;

default:  exit (0);

 }

}

}
```

Seen