Objective:

Write a program to implement a **Stack Operations** using a Array as a Stack.

Code :

```c
#include <stdio.h>
#define MAXQ 2
typedef struct stack {
    int A[MAXQ] ;
    int top;
} stack ;

void inserts( stack* , int);
int deletes( stack* );
void displays( stack);
void initialize( stack* );

int main(){
    stack s ;
    int ch , n ;
    initialize(&s);

    printf("1. Push \n");
    printf("2. Pop \n");
    printf("3. Display \n");
    printf("4. End \n");

    do{
        printf("Enter Choice : ");
        scanf("%d" , &ch );
        switch(ch) {
            case 1 :
            printf("Enter Value to Insert : ");
            scanf("%d" , &n);
            inserts(&s , n);
            break;
            case 2 :
            n = deletes(&s);
            if(n == -1) break;
            printf("Deleted Value : %d\n" , n );
            break;
            case 3 :
            displays(s);
            break;
        }

    } while ( ch != 4) ;
}

void initialize(stack *S){
    S->top = -1;
}
```

```c
void inserts( stack *S , int x ){

    if( S->top == MAXQ -1 ){
        printf("Stack is Full \n");
        return;
    }
    S->A[++S->top] = x ;
}

int deletes( stack *S ){

    int x ;
    if( S->top == -1 ){
        printf("Stack is Empty \n");
        return(-1);
    }
    x = S->A[S->top--] ;
    return(x);

}

void displays( stack S ){
    printf("Top -> ");
    for( int i = S.top ; i >= 0 ; i--){
        printf("%d " , S.A[i] );
    }
    printf("\n");
}
```

Output :

```
PS D:\College\DS\Stack> .\stack
1. Push
2. Pop
3. Display
4. End
Enter Choice : 1
Enter Value to Insert : 12
Enter Choice : 1
Enter Value to Insert : 13
Enter Choice : 1
Enter Value to Insert : 14
Stack is Full
Enter Choice : 3
Top -> 13 12
Enter Choice : 2
Deleted Value : 13
Enter Choice : 2
Deleted Value : 12
Enter Choice : 2
Stack is Empty
Enter Choice : 4
PS D:\College\DS\Stack>
```