

Program to illustrate operations on Array Stack

```
#include<stdio.h>
#include<conio.h>
#define SIZE 100
typedef struct
{
int elem[SIZE];
int top;
}stack;
void push(stack*,int);
int pop(stack*);
int underflow(stack);
int overflow(stack);
int peek(stack*);
void push(stack *s1,int info)
{
s1->elem[++s1->top]=info;
}
int pop(stack *s1)
{
return s1->elem[s1->top--];
}
int underflow(stack s1)
{
```

```
if(s1.top== -1)
return 1;
else
return 0;
}
int overflow(stack s1)
{
if(s1.top==SIZE)
return 1;
else
return 0;
}
int peek(stack *s1)
{
return s1->elem[s1->top];
}

void main()
{
stack s1;
int ch,data;
s1.top=-1;/*created empty stack*/
do
{
```

```
printf("\nFollowing operations can be performed on the stack");
printf("\n1.\tPush an element at top");
printf("\n2.\tPop an element from top");
printf("\n3.\tpeek an element at top");
printf("\n4.\tExit");
printf("\nEnter your choice ");
scanf("%i",&ch);
switch(ch)
{
case 1: if(overflow(s1))
        {
        printf("\nstack is overflown, pop some element and then try");
        break;
        }
        printf("\nEnter the data to store as information in stack: ");
        scanf("%i",&data);
        push(&s1,data);
        break;
case 2: if(underflow(s1))
        {
        printf("\n stack does not contain element, push some ");
        break;
        }
        data=pop(&s1);
```

```
        printf("\n%d deleted from stack top",data);
        break;
case 3: if(underflow(s1))
        {
        printf("\n stack does not contain element, push some ");
        break;
        }
        data=peek(&s1);
        printf("\n%d found at stack top",data);
        break;
case 4: return;
default: printf("\nTry 1 -4");
        break;
}
}while(ch!=4);
}
```