1) WAP to simulate the working of stack using an array with a) push, b) pop c) display. the program should print appropriate message for stack overflow and stack underflow

Soln:-

```
# include <stdio.h>
# include <stdlib.h>
# define STACK_SIZE 3
int s[3], item;
int top = -1;
void push ()
{
    if (top == STACK_SIZE - 1)
    {
        printf ("stack overflow \n");
    }
    top = top + 1;
    s[top] = item;
}
int pop ()
{
    if (top == -1) {
        printf ("stack underflow \n");
    }
    return s[ top = top - 1];
}
void display ()
{
    int i;
    if (top == -1) {
```

```
        printf ("Empty stack\n");
    }
    printf ("contents of the stack are\n");
    for (i=0; i<=top; i++){
        printf ("%d\n", s[i]);
    }
}
void main ()
{
    int item_deleted;
    int choice = 1;
    while (choice != 0)
    {
        printf ("1. push\n 2. pop\n 3. display\n 4. Exit\n");
        printf ("Enter choice\n");
        scanf ("%d", &choice);
        switch (choice) {
        case 1 : printf ("Enter number to be inserted\n");
        scanf ("%d", &item);
        push ();
        break;
        case 2: item_deleted = pop();
        if (item_deleted != 0) {
            printf ("item deleted is %d\n", item_deleted);
            break;
        }
        else {
            printf ("stack underflow\n");
        }
        break;
```

```
        case 3: display();
        break;
        case 4: choice = 0;
        break;
        default: printf("Invalid Input\n");
    }
    }
}
```

DS - Lab

[Passing Parameters]

1) WAP to simulate the working of stack using an array with a) Push b) Pop c) display. The program should print appropriate message for stack overflow and stack underflow.

Sol^n:-

```
#include <stdio.h>
#include <conio.h>
#include <process.h>
#define STACK_SIZE 5
int top = -1
void push (int item, int s[], int *top)
{
    if (*top == STACK_SIZE - 1)
    {
        printf ("Stack overflow \n");
    }
    *top = *top + 1;
    s[*top] = item;
}
int pop (int s[], int *top)
{
    int item_deleted;
    if (*top == -1)
    {
        printf ("Stack underflow \n");
    }
```

```
item_deleted = s[*top];
*top = *top -1;
return item_deleted;
}
int display (int top, int s[])
{
    int i;
    if (top == -1)
    {
        printf ("It is an empty stack\n");
        return 0;
    }
    printf ("The stack is \n");
    for (i=0; i <= top ; i++)
    {
        printf ("%d \n", s[i]);
    }
    return 0;
}
void main ()
{
    int item, s[5];
    int item_deleted;
    int choice = 1;
    while (choice != 0)
    {
        printf (" 1. push \n 2. pop \n 3. display \n 4. exit
        printf (" Enter number \n");
        scanf ("%d", &choice);
```

```
switch (choice)
{
    case 1 :    printf (" Enter number to be inserted \n");
                scanf (" %d ", & item );
                push ( item, S[], & top );
                break;

    case 2 :    item - deleted = pop ( S, & top );
                if ( item - deleted != 0 ) {
                printf (" item deleted is %d \n", item - deleted);
                break;

    case 3 :    display ( top, S);
                break;

    default : exit (0)
    }
    }
    getch ();
}

    case 4 :    choice = 0;
    break;
    case 5:
    default :   printf (" invalid input \n");
}
}
}
```