

Lab 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 and stack underflow.

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

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

```
#define stack_size 3
```

```
int top = -1
```

```
int s[3];
```

```
int item;
```

```
void push() {
```

```
    if (top == stack_size - 1)
```

```
        printf("stack overflow\n");
```

```
        return;
```

```
    }
```

```
    top = top + 1;
```

```
    s[top] = item;
```

```
}
```

```
int pop()
```

```
{
```

```
    if (top == -1) return -1;
```

```
    return s[top--];
```

```
}
```

```
void display ()
```

```
{  
    int i;
```

```
    if (top == -1)
```

```
    {  
        printf("stack is empty \n");  
        return;  
    }
```

```
    printf("contents of the stack \n");
```

```
    for (i = 0; i <= top; i++)
```

```
    {  
        printf("%d \n", s[i]);  
    }
```

```
void main()
```

```
{
```

```
    int item - deleted;
```

```
    int choice;
```

```
    for (i = 0;
```

```
    {  
        printf("\n 1: push \n 2: pop \n 3: display \n  
        4: exit \n");
```

```
        printf("enter the choice \n");
```

```
        scanf("%d", &choice);
```

```
        switch (choice)
```

```
        {  
            Case 1:
```

```
            printf("enter the item to be inserted \n");
```

```
            scanf("%d", &item);
```

```
            push();
```

```
            break;
```




Case 2 :

```
item_deleted = pop();
if (item_deleted == -1)
printf("stack is empty \n");
else
printf("Item deleted is %d\n", item_deleted);
break;
```

Case 3 :

```
display();
break;
default : exit(0);
}
```

```
{
}
```

OUTPUT :-

1: push

2: pop

3: display

4: exit

enter the choice

1

enter the item to be inserted

34

1: push

2: pop

3: display

4: exit

enter the choice

1

enter the item to be inserted
45

- 1: push
- 2: pop
- 3: display
- 4: exit

enter the choice

1

enter the element to be inserted
38

- 1: push
- 2: pop
- 3: display
- 4: exit

enter the choice

1

enter the item to be inserted
48

stack overflow

- 1: push
- 2: pop
- 3: display
- 4: exit

enter the choice

2

Item Deleted is 38

Date ____ / ____ / ____

1: push

2: pop

3: display

4: exit ▽

enter the choice

3

content of the stack

34

35

1: push

2: pop

3: display

4: exit ▽

enter the choice

4

Q2) Write a C program to simulate the working of stack using array ~~by passing arr~~ with push, pop and display.

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

void display (int s[])
{
    int i;
```

```
if (top == -1)
```

```
{
    printf("stack is empty \n");
    return;
}
```

```
printf("contents of the stack \n");
```

```
for (i = 0; i <= top; i++)
```

```
{
    printf("%d \n", s[i]);
}
```

```
}
```

```
void main()
```

```
{
```

```
    int s[3], item;
```

```
    int item_deleted;
```

```
    int choice;
```

```
    for (i = 0;
```

```
        printf("\n 1: push \n 2: pop \n 3: display \n 4: exit \n");
```

```
        printf("enter the choice: \n");
```

```
        scanf("%d", &choice);
```

```
        switch (choice)
```

```
        {
            case 1:
```

```
                printf("enter the item to be inserted \n");
```

```
                scanf("%d", &item);
```

```
                push(item, s);
```

```
                break;
```

```
            case 2:
```

```
                item_deleted = pop(s);
```


Date ___/___/___

```
if (item_deleted == -1)
    printf("stack is empty \n");
else
    printf("Item Deleted is '%d' \n",
           item_deleted);
break;
Case 3:
display(s);
break;
default: exit(0);
}
```

}