

Week-2

2) WAP to simulate the working of stack using an array with appropriate messages for stack overflow and stack underflow

```
→ #include <stdio.h>
#include <process.h>
#include <conio.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");
        return;
    }
    *top = *top + 1;
    S[*top] = item;
}
int pop(int S[], int *top)
{
    int deleted_item;
    if (*top == -1)
    {
        printf("Stack underflow Cannot delete\n");
        return 0;
    }
    deleted_item = S[*top];
    *top = *top - 1;
    return deleted_item;
}
```

Page _____

```
void display ( int top, int s[] )
```

```
{
    int i;
    if (top == -1)
    {
        printf ("Stack is empty \n");
        return ;
    }
```

```
    printf ("Contents of stack \n");
    for (i=0; i<=top, i++)
```

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

```
void main()
```

```
{
    int item, s[10];
    int deleted_item;
    int choice;
```

```
clrscr();
    for(;;)
```

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

```
    printf ("Enter your choice \n");
```

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

```
    switch (choice)
```

```
{
    case 1: printf ("Enter the item to be inserted \n");
            scanf scanf ("%d", &item);
            push (item, s, &top);
            break;
```

```
Case 2: deleted_item = pop(s, & top);  
        if (deleted_item != 0)  
            printf("\nitem deleted is %d \n", deleted_item);  
        break;
```

```
Case 3: display(top, s)  
        break;
```

```
default : exit(0)
```

```
}
```

```
}  
getch();  
}
```