

Week 1

Q) WAP to simulate the working of stack using an array with  
1) Push 2) Pop 3) Peek 3) Display  
functions

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

```
#define MAX 5
```

```
char stack[MAX];
```

```
int top = -1;
```

```
void push (char x) {
```

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

```
        printf("stack overflow");  
        return;
```

```
    }
```

```
    stack[++top] = x;
```

```
}
```

```
void peek () {
```

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

```
        printf("stack underflow");
```

```
        printf("top element : %c", stack[top]);
```

```
    }
```

```
void pop() {
```

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

```
printf("Stack underflow\n");
```

```
printf("popped element is : %d",
```

```
stack[top--]);
```

```
}
```

```
void display() {
```

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

```
printf("Stack Empty\n");
```

```
return;
```

```
}
```

```
printf("Stack contents:");
```

```
for (int i = top; i >= 0; i--) {
```

```
printf("%d ", stack[i]);
```

```
}
```

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

```
}
```

```
int main() {
```

```
int ops;
```

```
char x;
```

```
while(1) {
```

```
printf("\n 1 for push\n 2 for pop\n 3 for peek\n 4 for display");
```

```
printf("Enter operation : ");
```

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

switch(cops) {

case 1:

printf("Enter element:");  
scanf("%c", &x);  
push(x);  
break;

case 2:

pop();  
break;

case 3:

display();  
break;

case 4:

peek();  
break;

case 0:

printf("Existing -- \n");  
return 0;

default:

printf("Invalid operation \n");

}

}

}

14/10/24

Stack out:-

1 for push

2 for pop

3 for display

4 for peek

0 to exit

Enter op: 2

Stack underflow

Enter op: 3

Stack is empty

Enter op: 4

Stack underflow

Enter op: 1

Enter element 1

Enter op: 1

Enter element 1

Enter op: 1

Enter element 1

~~Enter op: 1~~

~~Enter element 1~~

Enter op: 1

Enter element 1

Stack overflow.