

### 3.Stack Implementation

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

```
int stack[100],i,n,ch=0,top=-1;
```

```
void push();
```

```
void pop();
```

```
void display();
```

```
void main()
```

 $\{$ 

```
while(ch!=4)
```

 $\{$ 

```
printf("Enter the size of the stack\n");
```

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

```
printf("Press:\n1:push()\n2:pop()\n3:display()\n4:Exit\n");
```

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

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

```
switch(ch)
```

 $\{$ 

case 1: push();

```
break;
```

case 2: pop();

```
break;
```

```
case 3:display();
```

```
break;
```

```
case 4:printf("Exiting...");
```

```
break;
```

```
default : printf("Press valid choice");
```

```
}
```

```
}
```

```
}
```

```
void push()
```

```
{
```

```
if(top==n)
```

```
{
```

```
printf("The stack is full\n");
```

```
}
```

```
else
```

```
{
```

```
int value;
```

```
printf("Enter the value\n");
```

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

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

```
stack[top]=value;
```

```
}
```

```
}
```

```
void pop()
```

```
{
    if(top==-1)
    {
        printf("Underflow");
    }
    else
    {
        top=top-1;
    }
}

void display()
{
    if(top==-1)
    {
        printf("The stack is empty");
    }
    else
    {
        for(i=top;i>=0;i--)
            printf("%d\n",stack[i]);
    }
}
```

## OUTPUT:

```
Enter the size of the stack
5
Press: 1:push() 2:pop() 3:display() 4:Exit
Enter the choice
1
Enter the value
6
Enter the size of the stack
5
Press: 1:push() 2:pop() 3:display() 4:Exit
Enter the choice
2
Enter the size of the stack
5
Press: 1:push() 2:pop() 3:display() 4:Exit
Enter the choice
3
The stack is emptyEnter the size of the stack
4
Press: 1:push() 2:pop() 3:display() 4:Exit
Enter the choice
4
Exiting...

...Program finished with exit code 4
Press ENTER to exit console.
```