

Write a C program to implement Stack operations such as PUSH, POP and PEEK

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

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

```
#define Size 4
```

```
int Top=-1, inp_array[Size];
```

```
void Push();
```

```
void Pop();
```

```
void show();
```

```
int main()
```

```
{
```

```
    int choice;
```

```
    while(1)
```

```
    {
```

```
        printf("\nOperations performed by Stack");
```

```
        printf("\n1.Push the element\n2.Pop the element\n3.Show\n4.End");
```

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

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

```
        switch(choice)
```

```
        {
```

```
            case 1: Push();
```

```
                    break;
```

```
            case 2: Pop();
```

```
                    break;
```

```
            case 3: show();
```

```
                    break;
```

```

        case 4: exit(0);

        default: printf("\nInvalid choice!!");
    }
}

void Push()
{
    int x;

    if(Top==Size-1)
    {
        printf("\nOverflow!!");
    }
    else
    {
        printf("\nEnter element to be inserted to the stack:");
        scanf("%d",&x);
        Top=Top+1;
        inp_array[Top]=x;
    }
}

void Pop()
{
    if(Top==-1)
    {
        printf("\nUnderflow!!");
    }
    else

```

```
{  
    printf("\nPopped element: %d",inp_array[Top]);  
    Top=Top-1;  
}  
}
```

```
void show()
```

```
{  
  
    if(Top== -1)  
    {  
        printf("\nUnderflow!!");  
    }  
    else  
    {  
        printf("\nElements present in the stack: \n");  
        for(int i=Top;i>=0;--i)  
            printf("%d\n",inp_array[i]);  
    }  
}
```

Operations performed by Stack

- 1.Push the element
- 2.Pop the element
- 3.Show
- 4.End

Enter the choice:1

Enter element to be inserted to the stack:10

Operations performed by Stack

- 1.Push the element
- 2.Pop the element
- 3.Show
- 4.End

Enter the choice:4

...Program finished with exit code 0

Press ENTER to exit console.