



1. Write a menu driven program in C to perform Stack operations (Push, Pop, Peek, Display) using user defined functions.

Program: prg1.c

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 4
void display();
void push();
void pop();
void peek();
int Stack[MAX], top = -1;
int main()
{
   int option;
   while (1)
       printf("\n\t\t-----\n");
       printf(" 0. Exit\n 1. PUSH\n 2. POP\n 3. PEEK\n 4. DISPLAY\n");
       printf("\nEnter a option : ");
       scanf("%d", &option);
       switch (option)
       {
       case 0:
           printf("\n\tTHANK YOU\n");
           exit(0);
       case 1:
           push();
           break;
       case 2:
           pop();
           break;
       case 3:
           peek();
           break;
       case 4:
           display();
           break;
```





```
void display()
    int r;
    if (top == -1)
        printf("\n\tStack is empty..!\n\n");
        return;
    }
    printf("\nThe STACK is : \n");
    for (r = top; r >= 0; r--)
        printf("\n\t%d", Stack[r]);
        if (r == top)
            printf(" <---- [top]");
    }
}
void push()
    int i, item;
    if (top == MAX - 1)
        printf("\n\tStack is full...!\n\n");
        return;
    printf("Enter new element : ");
    scanf("%d", &item);
    top = top + 1;
    Stack[top] = item;
    printf("\n Successfully pushed %d in the stack \n", item);
    display();
}
void pop()
{
    int k;
    if (top == -1)
        printf("\n Stack is empty..!\n\n");
        return;
    top = top - 1;
```





```
printf("\n The element %d is successfully deleted from top \n", Stack[top + 1]);
    display();
}

void peek()
{
    if (top == -1)
        {
        printf("\n Stack is empty...!\n\n");
        return;
        }
        printf("\n\n Top of the stack = %d at index = %d \n", Stack[top], top);
}
```





OUTPUT:

	STACK OPERATIONS	Successfully
	0. Exit 1. PUSH	The STACK is
	2. POP 3. PEEK	
	4. DISPLAY	
	Enter a option: 1	Press Enter to
	Enter new element : 1	Tress Enter to
	Successfully pushed 1 in the stack	0. Exit 1. PUSH
	The STACK is:	2. POP 3. PEEK
	1 < [top]	4. DISPLAY
	Press Enter to continue	Enter a option
	STACK OPERATIONS	Enter new eler
	0. Exit 1. PUSH	Successfully
	2. POP 3. PEEK	The STACK is
	4. DISPLAY	
	Enter a option: 1	3
	Enter new element : 2	
	Successfully pushed 2 in the stack	Press Enter to
	The STACK is:	0. Exit
	2 < [top] 1	1. PUSH
		2. POP 3. PEEK
	Press Enter to continue	4. DISPLAY
	STACK OPERATIONS 0. Exit	Enter a option
	1. PUSH 2. POP	Stac
	3. PEEK	
	4. DISPLAY	Press Enter to
	Enter a option: 1	0. Exit 1. PUSH
	Enter new element : 3	2. POP
ш	,	1 1 2 DEEE

Successfully pushed 3 in the stack		
The STACK is:		
3 < [top] 2 1		
Press Enter to continue		
STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY		
Enter a option: 1		
Enter new element : 4		
Successfully pushed 4 in the stack		
The STACK is:		
4 < [top] 3 2 1		
Press Enter to continue		
STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY		
Enter a option: 1		
Stack is full!		
Press Enter to continue STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY		



Programming for Problem Solving Lab (C) (ES-CS 291)

CSE - 1st Year - 2nd Sem.



A STORY OF THE STO		
Enter a option : 2		
The element 4 is successfully deleted from top		
The STACK is:		
3 < [top] 2 1		
Press Enter to continue		
Enter a option: 3		
Top of the stack = 3 at index = 2		
Press Enter to continue STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY		
Enter a option : 4		
The STACK is:		
3 < [top] 2 1		
Press Enter to continue		
STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY		
Enter a option : 2		
The element 3 is successfully deleted from top		
The STACK is:		
2 < [top]		

```
Press Enter to continue....
        ----- STACK OPERATIONS -----
    0. Exit
    1. PUSH
    2. POP
    3. PEEK
    4. DISPLAY
Enter a option: 2
          The element 2 is successfully deleted from top
The STACK is:
             | 1 | <---- [top]
Press Enter to continue....
        ----- STACK OPERATIONS -----
    0. Exit
    1. PUSH
    2. POP
    3. PEEK
    4. DISPLAY
Enter a option: 2
          The element 1 is successfully deleted from top
The STACK is:
             Stack is empty..!
Press Enter to continue....
       ----- STACK OPERATIONS -----
    0. Exit
    1. PUSH
    2. POP
    3. PEEK
    4. DISPLAY
Enter a option: 2
          Stack is empty..!
Press Enter to continue....
        ----- STACK OPERATIONS -----
    0. Exit
    1. PUSH
    2. POP
    3. PEEK
    4. DISPLAY
```





Enter a option: 0		
THANK YOU		





2. Write a menu driven program in C to perform Stack operations (Push, Pop, Peek, Display) using Structure data type.

Program: prg2.c

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 4
struct stack
{
   int top;
   int ar[MAX];
} stk;
void push();
void pop();
void peek();
void display();
int main()
{
   int option;
   system("cls");
   stk.top = -1;
   while (1)
       system("cls");
       printf("\n\t\t-----\n");
       printf(" 0. Exit\n 1. PUSH\n 2. POP\n 3. PEEK\n 4. DISPLAY\n");
       printf("\nEnter your choice : ");
       scanf("%d", &option);
       switch (option)
       case 0:
           printf("\n\tTHANK YOU\n");
           exit(0);
       case 1:
           push();
           display();
           break;
```





```
case 2:
            printf("\nPopped Elements.\n");
            pop();
            display();
            break;
        case 3:
            peek();
            break;
        case 4:
            display();
            break;
        default:
            printf("\n\tERROR.. Wrong Choice !!!\t");
            break;
        fflush(stdin);
        getchar();
    }
    return 0;
}
void push()
    if (stk.top == MAX - 1)
        printf("\nStack Overflow !");
    else
        printf("\nEnter new element : ");
        scanf("%d", &stk.ar[++stk.top]);
    }
}
void pop()
    if (stk.top == -1)
        return;
    else
        printf("\nThe %d Element is Popped.\n", stk.ar[stk.top--]);
}
```





```
void peek()
    if (stk.top == -1)
        printf("\nStack Underflow!");
    else
    {
        printf("\nTop Elements is: %d\n", stk.ar[stk.top]);
}
void display()
{
    int i;
    if (stk.top < 0)
        printf("\nStack Underflow!");
    else
        printf("\n\nThe Stack is: \n");
        for (i = stk.top; i >= 0; i--)
        {
            printf("\n| %d |", stk.ar[i]);
            if (i == stk.top)
                printf(" <--- [Top(%d)]", stk.top);</pre>
        }
    }
}
```





OUTPUT:

STACK OPERATIONS	STACK OPERATIONS -
0. Exit	0. Exit
1. PUSH	1. PUSH
2. POP	2. POP
3. PEEK	3. PEEK
4. DISPLAY	4. DISPLAY
II DIGITALIT	" BISTERT
Enter your choice: 1	Enter your choice : 1
2.1101 y cm. 4.16100 v 1	Zanor year energe vi
Enter new element : 1	Enter new element : 4
The STACK is:	The STACK is:
14.1	57 (2)
1 < [Top(0)]	4 < [Top(3)] 3
	3
Press Enter to continue	2
STACK OPERATIONS	1 * 1
	D E 4 4 4
0. Exit	Press Enter to continue
1. PUSH	
2. POP	STACK OPERATIONS -
3. PEEK	0. Exit
4. DISPLAY	1. PUSH
4. DISI LA I	
	2. POP
Enter your choice: 1	3. PEEK
	4. DISPLAY
Enter new element : 2	
Enter new element . 2	Enter your choice: 1
TI OTACK:	Effect your choice . I
The STACK is:	
	Enter new element : 5
2 < [Top(1)] 1	
j1j	Stack Overflow!
1-1	
Press Enter to continue	The STACK is:
riess Enter to continue	THE STACK IS.
STACK OPERATIONS	4 < [Top(3)] 3
0. Exit	3
1. PUSH	
	' '
2. POP	1
3. PEEK	
4. DISPLAY	Press Enter to continue
Enter your choice: 1	STACK OPERATIONS -
	0. Exit
Enter new element: 3	1. PUSH
Litter new ciclient. J	
TIL OTHER CITE	2. POP
The STACK is:	3. PEEK
	4. DISPLAY
3 < [Top(2)]	
	Enter your aboles : 2
2	Enter your choice : 2
1	
Press Enter to continue	The 4 Element is Popped.





The STACK is:	Press Enter to continue
3 < [Top(2)] 2 1 Press Enter to continue	STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK
STACK OPERATIONS	4. DISPLAY
0. Exit 1. PUSH 2. POP	Enter your choice : 2
3. PEEK 4. DISPLAY	The 2 Element is Popped. The STACK is:
Enter your choice : 3	1 < [Top(0)]
Top Elements is: 3	Press Enter to continue
Press Enter to continue	
STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY	STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY
Enter your choice : 4	Enter your choice: 2
	The 1 Element is Popped.
The STACK is:	The STACK is:
3 < [Top(2)] 2 1	Stack Underflow! Press Enter to continue
Press Enter to continue	
STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY	STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY
Enter your choice : 2	Enter your choice : 2
The 3 Element is Popped.	Stack Underflow!
The STACK is:	Press Enter to continue
2 < [Top(1)] 1	STACK OPERATIONS 0. Exit 1 PUSH





2. POP

3. PEEK

4. DISPLAY

Enter your choice: 0

THANK YOU





3. Write a menu driven program in C to perform Stack operations (Push, Pop, Peek, Display) using Structure Pointer .

Program: prg3.c

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 4
struct stack
{
   int top;
   int ar[MAX];
} *stk;
void push();
void pop();
void peek();
void display();
int main()
{
   int option;
   system("cls");
   stk = (struct stack *)malloc(sizeof(struct stack));
   stk->top = -1;
   while (1)
       system("cls");
       printf("\n\t\t-----\n");
       printf(" 0. Exit\n 1. PUSH\n 2. POP\n 3. PEEK\n 4. DISPLAY\n");
       printf("\nEnter your choice : ");
       scanf("%d", &option);
       switch (option)
       case 0:
           printf("\n\tTHANK YOU\n");
           exit(0);
       case 1:
           push();
           display();
           break;
```





```
case 2:
            printf("\nPopped Elements.\n");
            pop();
            display();
            break;
        case 3:
            peek();
            break;
        case 4:
            display();
            break;
        default:
            printf("\n\tERROR.. Wrong Choice !!!\t");
            break;
        fflush(stdin);
        getchar();
    }
    return 0;
}
void push()
    if (stk->top == MAX - 1)
        printf("\nStack Overflow !");
    else
        printf("\nEnter new element : ");
        scanf("%d", &stk->ar[++stk->top]);
    }
}
void pop()
    if (stk->top == -1)
        return;
    else
        printf("\nThe %d Element is Popped.\n", stk->ar[stk->top--]);
}
```





```
void peek()
    if (stk->top == -1)
        printf("\nStack Underflow!");
    else
    {
        printf("\nTop Elements is: %d\n", stk->ar[stk->top]);
}
void display()
{
    int i;
    if (stk->top < 0)
        printf("\nStack Underflow!");
    else
        printf("\n\nThe Stack is: \n");
        for (i = stk->top; i >= 0; i--)
        {
            printf("\n| %d |", stk->ar[i]);
            if (i == stk->top)
                printf(" <--- [Top(%d)]", stk->top);
        }
    }
}
```



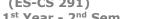


OUTPUT:

STACK OPERATIONS 0. Exit	
1. PUSH 2. POP 3. PEEK 4. DISPLAY	
Enter your choice : 1	
Enter new element : 1	
The STACK is:	
1 < [Top(0)]	
Press Enter to continue	
STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY	
Enter your choice : 1	
Enter new element : 2	
The STACK is:	
2 < [Top(1)] 1	
Press Enter to continue	
STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY	
Enter your choice : 1	
Enter new element : 3	
The STACK is:	
3 < [Top(2)] 2 1 Press Enter to continue	

```
----- STACK OPERATIONS -----
0. Exit
1. PUSH
2. POP
3. PEEK
4. DISPLAY
Enter your choice: 1
Enter new element: 4
The STACK is:
         | 4 | < ---- [Top(3)]
         3
         | 2 |
         | 1 |
Press Enter to continue....
         ----- STACK OPERATIONS -----
0. Exit
1. PUSH
2. POP
3. PEEK
4. DISPLAY
Enter your choice: 1
Enter new element: 5
    Stack Overflow!
The STACK is:
         | 4 | < ---- [Top(3)]
         3 |
         | 2 |
         | 1 |
Press Enter to continue....
         ----- STACK OPERATIONS -----
0. Exit
1. PUSH
2. POP
3. PEEK
4. DISPLAY
Enter your choice: 2
    The 4 Element is Popped.
```







The STACK is:	Press Enter to continue
3 < [Top(2)] 2 1	STACK OPERATIONS 0. Exit 1. PUSH
Press Enter to continue	2. POP 3. PEEK
STACK OPERATIONS	4. DISPLAY
0. Exit 1. PUSH	Enter your choice : 2
2. POP 3. PEEK	The 2 Element is Popped.
4. DISPLAY	The STACK is:
Enter your choice: 3	1 < [Top(0)]
Top Elements is: 3	Press Enter to continue
Press Enter to continue	
STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK	STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY
4. DISPLAY	Enter your choice: 2
Enter your choice : 4	The 1 Element is Popped.
The STACK is:	The STACK is:
3 < [Top(2)] 2 1	Stack Underflow! Press Enter to continue
Press Enter to continue	
STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY	STACK OPERATIONS 0. Exit 1. PUSH 2. POP 3. PEEK 4. DISPLAY
Enter your choice : 2	Enter your choice : 2
The 3 Element is Popped.	Stack Underflow!
The STACK is:	Press Enter to continue
2 < [Top(1)] 1	STACK OPERATIONS 0. Exit 1. PUSH





2. POP

3. PEEK

4. DISPLAY

Enter your choice: 0

THANK YOU