Name: Vinayak Kumar Singh

Register No: 23MCA1030

Batch: 1 of MCA

Question: Stack Implementation using Array

Stack Code

stackinarray.c

stackinarray.c > pop()

```c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#define n 3

int stack[n];
int top = -1;
void push()
{
    if (top == n - 1)
    {
        printf("stack is full\n");
    }
    else
    {
        int x;
        printf("Enter value to be pushed into the stack \n");
        scanf("%d", &x);
        top = top + 1;
        stack[top] = x;
    }
}
int peek()
{
    int x = stack[top];
    printf("The value at top of stack is : %d\n", x);
    return x;
}
```

```
29  int pop()
30  {
31      int x = stack[top];
32      if (top == -1)
33      {
34          printf("Please enter any data in stack \n");
35      }
36      else
37      {
38          int x = stack[top];
39          printf("value poped is : %d \n", x);
40          top = top - 1;
41          return x;
42      }
43      return -1;
44  }
45  bool isfull()
46  {
47      if (top == n - 1)
48      {
49          printf("Stack is full \n");
50          return true;
51      }
52      else
53      {
54          printf("stack is not full\n");
55          return false;
56      }
57  }
```

```
58  bool isEmpty()
59  {
60      if (top == -1)
61      {
62          printf("Stack is empty \n");
63          return true;
64      }
65      else
66      {
67          printf("stack has some data\n");
68          return false;
69      }
70  }
71  int main()
72  {
73      printf("Stack Capacity is : %d\n", n);
74      int data;
75      while (1)
76      {
77          printf("Choose any of the following options:\n");
78          printf("Press 1 to Push \n");
79          printf("Press 2 to Peek \n");
80          printf("Press 3 to Pop \n");
81          printf("Press 4 to stack is full\n");
82          printf("Press 5 to stack is empty\n");
83          printf("Press 6 to Exit \n");
84          scanf("%d", &data);
85          switch (data)
86          {
87          case 1:
```

```
85          switch (data)
86          {
87          case 1:
88              push();
89              break;
90          case 2:
91              peek();
92              break;
93          case 3:
94              pop();
95              break;
96          case 4:
97              isfull();
98              break;
99          case 5:
100             isEmpty();
101             break;
102         case 6:
103             printf("Program Exit");
104             exit(0);
105             break;
106         default:
107             printf("Please Choose a value between 1 to 6\n");
108             break;
109         }
110     }
111     return 0;
112 }
```

Push operation in Stack

```
d:\MCA\2. Data Structures and Algorithms + LAB\DSA>cd "d:\MCA\2. Data Structures and Algorithms +
"stackinarray
Stack Capacity is : 3
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
1
Enter value to be pushed into the stack
11
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
1
Enter value to be pushed into the stack
22
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
1
Enter value to be pushed into the stack
33
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
1
stack is full
```

Peek and Pop into Stack

```
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
2
The value at top of stack is : 33
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
3
value poped is : 33
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
2
The value at top of stack is : 22
```

Checking if stack is full and empty

```
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
4
stack is not full
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
5
stack has some data
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
1
Enter value to be pushed into the stack
55
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
4
Stack is full
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
5
stack has some data
```

Exit the Program

```
Choose any of the following options:
Press 1 to Push
Press 2 to Peek
Press 3 to Pop
Press 4 to stack is full
Press 5 to stack is empty
Press 6 to Exit
6
Program Exit
d:\MCA\2. Data Structures and Algorithms + LAB\DSA>
```

END