Name-Jeevan R

Sec- I

DSA in C

Lab Program 1: Write a program to simulate the working of stack using an array with the following:

a) Push

b) Pop

c) Display

The program should print appropriate messages for stack overflow, stack underflow

Code:

```
#include <stdio.h>
#define SIZE 5

int stack[SIZE];

int top = -1;
// Function to push an element into the stack
void push(int value) {
   if (top == SIZE - 1) {
      printf("Stack Overflow! Cannot push %d\n", value);
   } else {
      top++;
      stack[top] = value;
      printf("%d pushed into stack\n", value);
   }
}

// Function to pop an element from the stack
void pop() {
    if (top == -1) {
        printf("Stack Underflow! No element to pop\n");
    } else {
        printf("%d popped from stack\n", stack[top]);
        top--;
    }
}

// Enhanced function to display stack elements
void display() {
    if (top == -1) {
        printf("Stack is empty!\n");
    } else {
        printf("\nCurrent Stack (Top to Bottom):\n");
        for (int i = top; i >= 0; i--) {
            if (i == top) {
                printf("| %d |  <- Top\n", stack[i]);
            } else {
                printf("| %d |\n", stack[i]);
            }
        }
        printf(" ----- \n");
    }
}

// Main function to run the menu-driven stack program
int main() {
    int choice, value;

    while (1) {
        printf("\n---- Stack Menu ----\n");
        printf("1. Push\n");
        printf("2. Pop\n");
        printf("3. Display Stack\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to push: ");
                scanf("%d", &value);
                push(value);
                break;
            case 2:
                pop();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting program.\n");
                return 0;
            default:
                printf("Invalid choice! Try again.\n");
        }
    }

    return 0;
}
```

Expected Output:

```
---- Stack Menu ----
```

```
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 3
Stack is empty!

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
Enter value to push: 10
10 pushed into stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
Enter value to push: 20
20 pushed into stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
Enter value to push: 30
30 pushed into stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
Enter value to push: 40
Stack Overflow! Cannot push 40

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 3

Current Stack (Top to Bottom):
| 30 |  <- Top
| 20 |
| 10 |
 -----

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
```

```
30 popped from stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
20 popped from stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
10 popped from stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
Stack Underflow! No element to pop

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
```

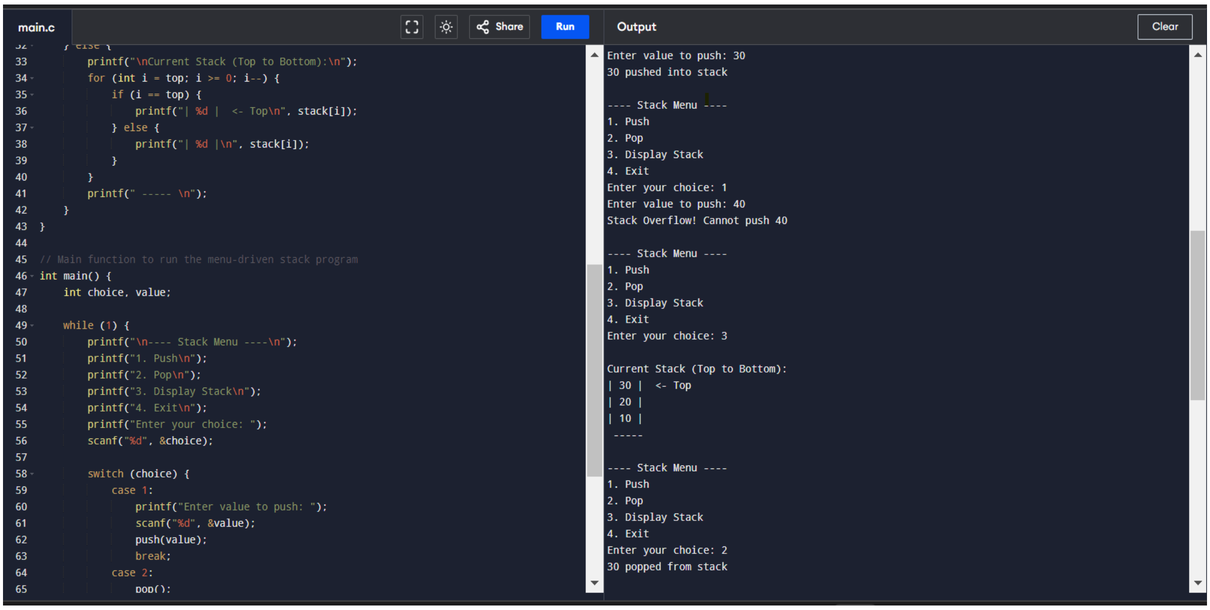
```
Enter your choice: 4
Exiting program.


=== Code Execution Successful ===---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 3
Stack is empty!

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
Enter value to push: 10
10 pushed into stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
Enter value to push: 20
20 pushed into stack

---- Stack Menu ----
```

```
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
Enter value to push: 30
30 pushed into stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
Enter value to push: 40
Stack Overflow! Cannot push 40

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 3

Current Stack (Top to Bottom):
| 30 |  <- Top
| 20 |
| 10 |
 -----

---- Stack Menu ----
```

```
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
30 popped from stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
20 popped from stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
10 popped from stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
Stack Underflow! No element to pop
```

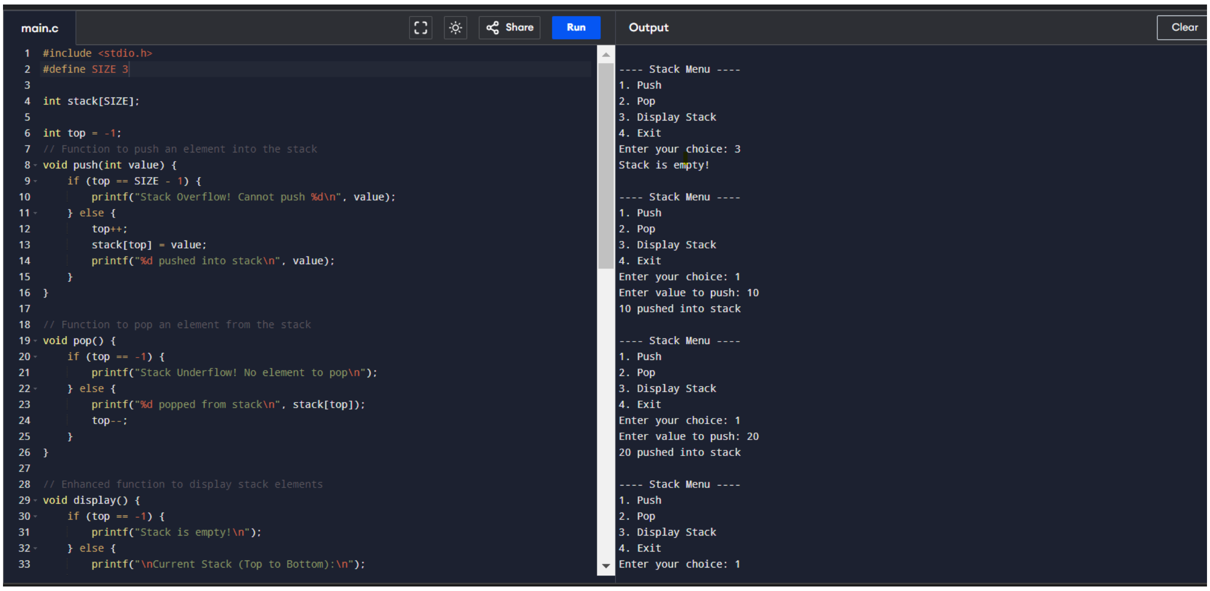
```
---- Stack Menu ----

1. Push

2. Pop

3. Display Stack

4. Exit

Enter your choice: 4

Exiting program.


=== Code Execution Successful ===
```

Execution Images:



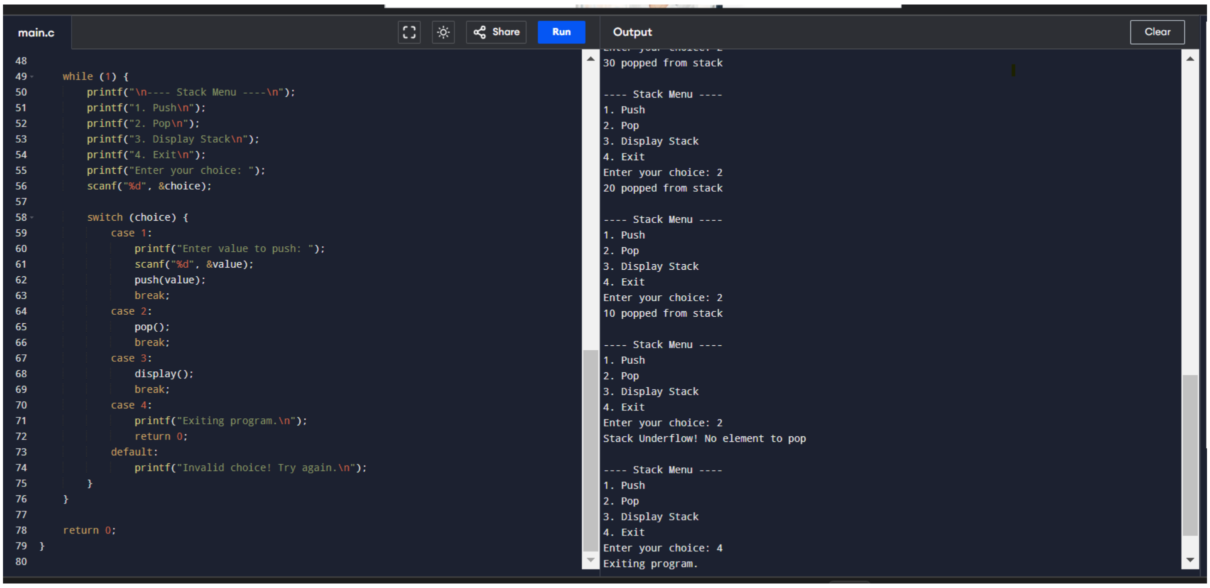
```c
    } else {
        printf("\nCurrent Stack (Top to Bottom):\n");
        for (int i = top; i >= 0; i--) {
            if (i == top) {
                printf("| %d |  <- Top\n", stack[i]);
            } else {
                printf("| %d |\n", stack[i]);
            }
        }
        printf(" ----- \n");
    }
}

// Main function to run the menu-driven stack program
int main() {
    int choice, value;

    while (1) {
        printf("\n---- Stack Menu ----\n");
        printf("1. Push\n");
        printf("2. Pop\n");
        printf("3. Display Stack\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to push: ");
                scanf("%d", &value);
                push(value);
                break;
            case 2:
                pop();
```

```
Enter value to push: 30
30 pushed into stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
Enter value to push: 40
Stack Overflow! Cannot push 40

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 3

Current Stack (Top to Bottom):
| 30 |  <- Top
| 20 |
| 10 |
 -----

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
30 popped from stack
```



```c
#include <stdio.h>
#define SIZE 3

int stack[SIZE];

int top = -1;
// Function to push an element into the stack
void push(int value) {
    if (top == SIZE - 1) {
        printf("Stack Overflow! Cannot push %d\n", value);
    } else {
        top++;
        stack[top] = value;
        printf("%d pushed into stack\n", value);
    }
}

// Function to pop an element from the stack
void pop() {
    if (top == -1) {
        printf("Stack Underflow! No element to pop\n");
    } else {
        printf("%d popped from stack\n", stack[top]);
        top--;
    }
}

// Enhanced function to display stack elements
void display() {
    if (top == -1) {
        printf("Stack is empty!\n");
    } else {
        printf("\nCurrent Stack (Top to Bottom):\n");
```

```

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 3
Stack is empty!

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
Enter value to push: 10
10 pushed into stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
Enter value to push: 20
20 pushed into stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 1
```



```c

    while (1) {
        printf("\n---- Stack Menu ----\n");
        printf("1. Push\n");
        printf("2. Pop\n");
        printf("3. Display Stack\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter value to push: ");
                scanf("%d", &value);
                push(value);
                break;
            case 2:
                pop();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting program.\n");
                return 0;
            default:
                printf("Invalid choice! Try again.\n");
        }
    }

    return 0;
}
```

```
Enter your choice: 2
30 popped from stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
20 popped from stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
10 popped from stack

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 2
Stack Underflow! No element to pop

---- Stack Menu ----
1. Push
2. Pop
3. Display Stack
4. Exit
Enter your choice: 4
Exiting program.
```