WEEK 01

Stack implementation using arrays :

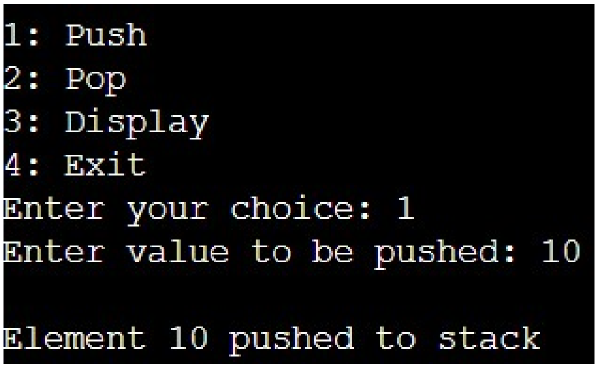
Code:

```
#include <stdio.h>

#include <stdlib.h>

#define SIZE 3


int top = -1;

int stack[SIZE];


void push(int item) {

    if (top == SIZE - 1) {

        printf("\nStack overflow");

    } else {

        top++;

        stack[top] = item;

        printf("\nElement %d pushed to stack", item);

    }

}


void pop() {

    if (top == -1) {

        printf("\nStack underflow");

    } else {
        printf("\nElement popped is %d", stack[top]);
        top--;
    }
}

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

int main() {
    int ch, item;
    for (;;) {
        printf("\n\n1: Push");
        printf("\n2: Pop");
        printf("\n3: Display");
        printf("\n4: Exit");
        printf("\nEnter your choice: ");
```

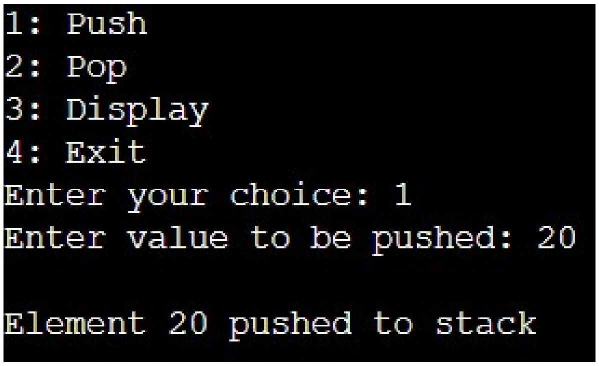
```
scanf("%d", &ch);

switch (ch) {
    case 1:
        printf("Enter value to be pushed: ");
        scanf("%d", &item);
        push(item);
        break;

    case 2:
        pop();
        break;

    case 3:
        display();
        break;

    case 4:
        exit(0);
        break;

    default:
        printf("\nInvalid choice. Please try again.");
        break;
        }
    }
    return 0;
}
```

Output :

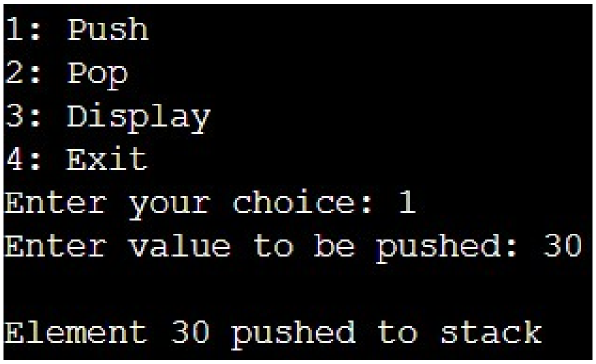
Pushing values:



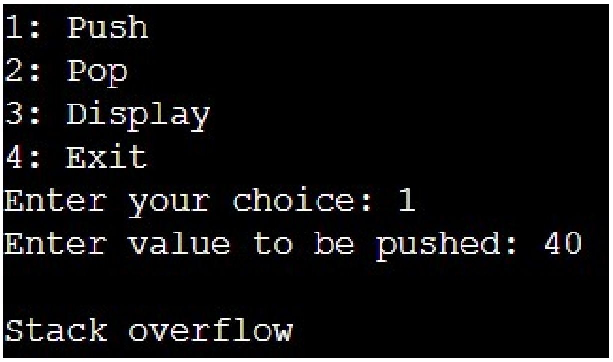
```
1: Push
2: Pop
3: Display
4: Exit
Enter your choice: 1
Enter value to be pushed: 10

Element 10 pushed to stack
```



```
1: Push
2: Pop
3: Display
4: Exit
Enter your choice: 1
Enter value to be pushed: 20

Element 20 pushed to stack
```

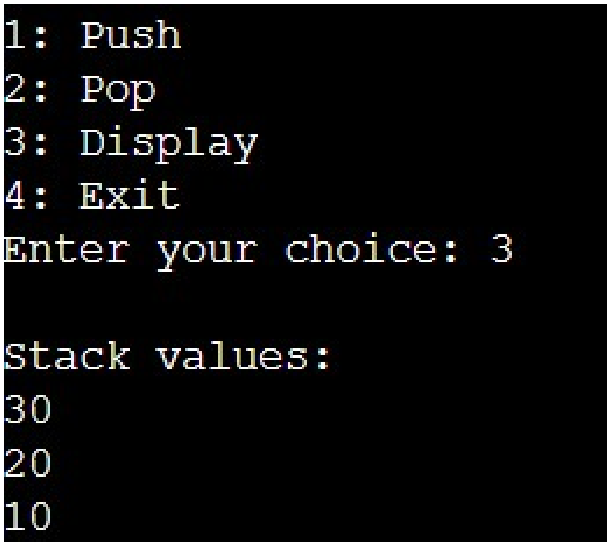


```
1: Push
2: Pop
3: Display
4: Exit
Enter your choice: 1
Enter value to be pushed: 30

Element 30 pushed to stack
```

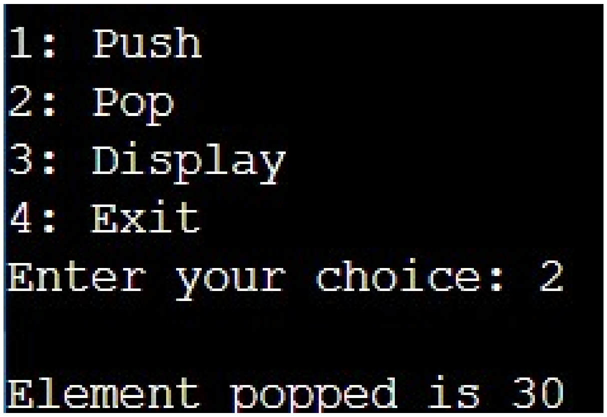


```
1: Push
2: Pop
3: Display
4: Exit
Enter your choice: 1
Enter value to be pushed: 40

Stack overflow
```

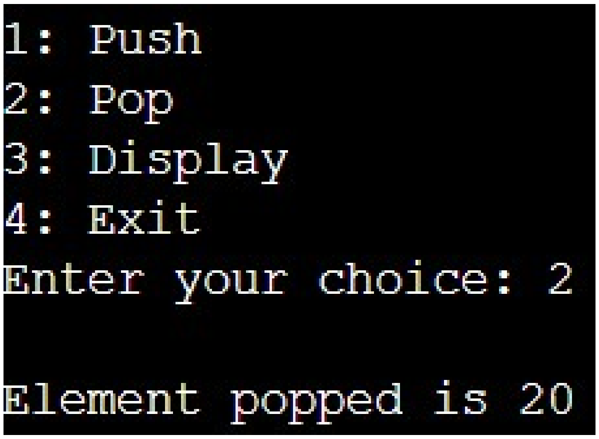
Poping values:



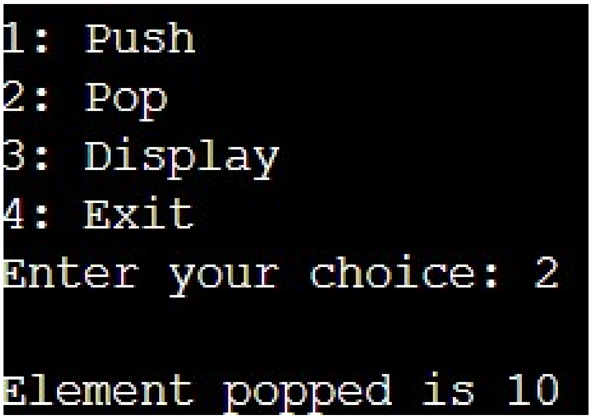
```
1: Push
2: Pop
3: Display
4: Exit
Enter your choice: 3

Stack values:
30
20
10
```



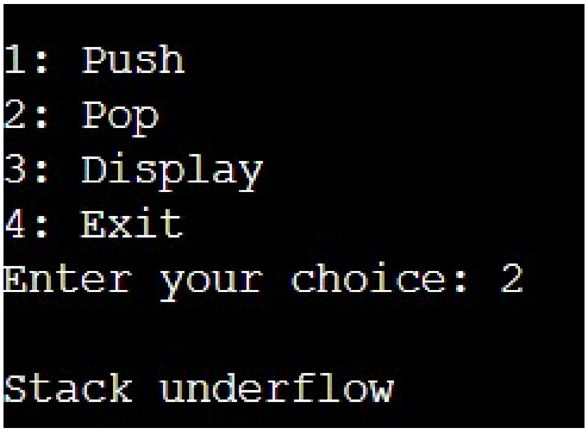
```
1: Push
2: Pop
3: Display
4: Exit
Enter your choice: 2

Element popped is 30
```



```
1: Push
2: Pop
3: Display
4: Exit
Enter your choice: 2

Element popped is 20
```

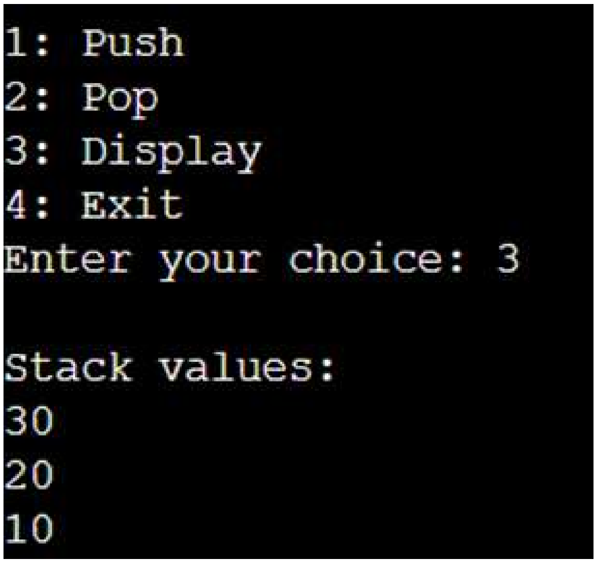


```
1: Push
2: Pop
3: Display
4: Exit
Enter your choice: 2

Element popped is 10
```

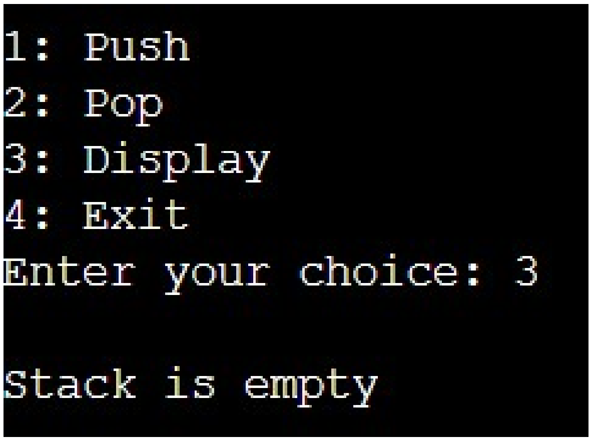


```
1: Push
2: Pop
3: Display
4: Exit
Enter your choice: 2

Stack underflow
```

Display:



```
1: Push
2: Pop
3: Display
4: Exit
Enter your choice: 3

Stack values:
30
20
10
```



```
1: Push
2: Pop
3: Display
4: Exit
Enter your choice: 3

Stack is empty
```

01/10/2024

WEEK 01

```
function push
    if ~~max~~ stack is full
        print "stack overflow"
    end if
    top = top + 1
    stack [ top ] = value
    print "valued pushed", value
end function.

function pop
    if stack is empty
    print "stack overflow"
    end if
    value = stack [top]
    top = top - 1
    print "value popped", value
    return value
end function

function display:
 If stack is empty then
   print "stack is empty"
   return
 endif
 print "stack elements"
 for i from top to down
    print stack[i]
 end for
endfunction.
```

```
# include <stdio.h>
# include <stdlib.h>
# define size 3

int top = -1;
int stack[size];

void push (int item) {
    if (top == size-1) {
        printf ("\n stack overflow");
    } else {
        top ++;
        stack [top] = item;
        printf ("\n element %d pushed to stack", item);
    }
}

void pop() {

    if (top == -1) {
        printf ("/n stack overflow");
    } else {
        printf ("\n element popped is %d", stack
                                            [top]);
        top --;
    }
}

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

```
int main (){
    int ch, item;
    for (;;){
      printf ("\n 1: Push");
      printf ("\n 2: Pop");
      printf ("\n 3: Display);
      printf ("\n 4: Exit);
      printf ("%d", & ch);
     switch (ch){
          Case 1;
             printf ("Enter Value to be pushed");
             scanf ("%d", & item);
             push (item);
             break;
         Case 2:
             ~~printf~~ pop();
                    break;
          Case 3:
              display ();
              break;
           Case 4:
              exit (0);
              break;
          default:
              printf ("\n invalid choice");
              break;
      }
   }
   return 0;
}
```

Output:

```
1: push
2: pop
3: display
4: exit
Enter your choice: 1
Enter value to be pushed: 10
Element 10 pushed to stack

1: push
2: pop
3: display
4: exit
Enter your choice: 1
Enter value to be pushed: 20
Element 20 pushed to stack

1: push
2: pop
3: display
4: exit
Enter your choice: 1
Enter value to be pushed: 30
Element 30 pushed to stack

1: push
2: pop
3: display
4: exit
Enter your choice: 3
Stack values: 30.
              20
              10
```

```
1: push
2: pop
3: display
4: exit
Enter your choice: 2
Element popped is 30

1: push
2: pop
3: display
4: exit
Enter your choice:1
element popped is 20

1: push
2: pop
3: display
4: exit
enter your choice: 3
Stack values : 10
```

1/10/24