## 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 \ge 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

4: Exit

3: Display

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 puch
   if mans stack is full
       peint "Stack overflow"
   end if
   top = top +1
   Stack [top] = value
   peint " valued pushed ", value
  end function.
 function pop
    if stack is empty
    peint "Stack overflow"
   end if
    value = stack [top]
   top = top-1
  peurt " value popped ", value
    Letuer value
   end functions
 function display:
  If stack is empty then
   print "spack is empty"
   Letwen
  endif
```

peird "stack elements" for i from top to down puint stack[i] end for endfunction

```
# undude < oddie ·h>
# unclude < stdlib.h>
# define size 3
int top = -1;
int stack (size];
void push (int item) {
     if ( top == size - 1) {
        purity ("In stack overflow");
     y else {
         dep ++;
         stack[70p]=item;
         fruith ("In element of a pushed to stack", item);
 void pop() {
       uf (top = = -1) {
          printly ("In stack overflow");
         y else &
              puints (" In element pepped is "/. d", stack
  void display () {
       if (top = = -1) {
         print ("In stace is empty");
        y else s
            puirdf (" / y stack values");
             for (int) = top; i>= 0; i--){
                printly ("In.1.d", stack [i]);
```

```
int main () {
     int uch, "talm;
    for (;;) {
      painty ("In 1: Push");
      fruints ("In 2: Pop");
       puintf (" (n 3: Display);
       printf ("In 4: Exit);
       printy (" /, d", & ch);
       suitch (ch) &
              prints ("Enter Value to be pushed");
              scenf (" /.d", & tem);
              push (item);
             break;
           Case 2:
              paints pop();
                 break;
            Case 3:
               display ();
                break &
             Case 4:
                enit (0);
                break;
             default:
                periody ("In invalid choice");
              Joreak;
   return 0;
```

Output:

1: puch

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: exil

Entu your choice: 1

Enter value to be pushed: 20

Element 20 pushed to stock

1: push

2: pop

3: display

4: exit

Enter your choice: 1

Enter value to be pushed: 30

Climent 30 pushed to stack

1: push

2: pap

3: display

4: buil

Enter your choice: 3

Stack values: 30.

50

10

1: bush

2: pop

3: display

4: enit

Enter your choice: 2 clement popped is 30

1: push

2: pop

3: display

4: enit

Enter your choice: 1 element popped is 20

1: puh

2: pop

3: display

4: enit choice: 3

Stack values: 10

Sedal B 24