

## EXPERIMENT-1

**Program 1: Design, Develop and Implement a menu driven Program in C for the following Array operations**

- a. Creating an Array of N Integer Elements**
- b. Display of Array Elements with Suitable Headings**
- c. Inserting an Element (ELEM) at a given valid Position (POS)**
- d. Deleting an Element at a given valid Position(POS)**
- e. Exit.**

**Support the program with functions for each of the above operations.**

### ABOUT THE EXPERIMENT:

- An Array is a collection of similar /same elements. In this experiment the array can be represented as one / single dimensional elements.
- Menu driven program in c - language to perform various array operations are implemented with the help of user defined functions as followings;
  - a. create() b. display() c. insert() d. del() e. exit()

### ALGORITHM:

Step 1: Start.

Step 2: Read N value.

Step 3: Read Array of N integer elements

Step 4: Print array of N integer elements.

Step 5: Insert an element at given valid position in an array.

Step 6: Delete an element at given valid position from an array.

Step 7: Stop.

### Program Code

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 5

int a[MAX], pos, elem;
int n = 0;

/*Function Prototype*/
void create();
void display();
void insert();
void delete();

void main()
{
    int choice;
```

```
while(1)
{
    printf("\n\n~~~~MENU~~~~");
    printf("\n=>1. Create an array of N integers");
    printf("\n=>2. Display of array elements");
    printf("\n=>3. Insert ELEM at a given POS");
    printf("\n=>4. Delete an element at a given POS");
    printf("\n=>5. Exit");
    printf("\n Enter your choice: ");
    scanf("%d", &choice);
    switch(choice)
    {
        case 1:      create();
                    break;
        case 2:      display();
                    break;
        case 3:      insert();
                    break;
        case 4:      delete();
                    break;
        case 5:      exit(1);
                    break;
        default:     printf("\nPlease enter a valid choice:");
    }
}
}
```

### **/\*Creating an Array\*/**

```
void create()
{
    int i;
    printf("\n Enter the number of elements: ");
    scanf("%d", &n);
    printf("\n Enter the elements: ");
    for(i=0; i<n; i++)
    {
        scanf("%d", &a[i]);
    }
}
```

### **/\*Displaying an array elements\*/**

```
void display()
{
    int i;
    if(n == 0)
```

```
{
    printf("\n No elements to display");
    return;
}
printf("\nArray elements are: ");
for(i=0; i<n;i++)
    printf("%d\t ", a[i]);
}
```

**/\*Inserting an element into an array\*/**

```
void insert()
{
    int i;

    if(n == MAX)
    {
        printf("\nArray is full. Insertion is not possible");
        return;
    }

    do
    {
        printf("\n Enter a valid position where element to be inserted:  ");
        scanf("%d", &pos);
    }while(pos > n);

    printf("\n Enter the value to be inserted:  ");
    scanf("%d", &elem);

    for(i=n-1; i>=pos ; i--)
    {
        a[i+1] = a[i];
    }
    a[pos] = elem;
    n = n+1;
    display();
}
```

**/\*Deleting an array element\*/**

```
void delete()
{
    int i;

    if(n == 0)
    {
        printf("\nArray is empty and no elements to delete");
        return;
    }
}
```

```
    }
    do
    {
        printf("\n Enter a valid position from where element to be deleted:  ");
        scanf("%d", &pos);
    }
    while(pos>=n);

    elem = a[pos];

    printf("\n Deleted element is : %d \n", elem);
    for( i = pos; i< n-1; i++)
    {
        a[i] = a[i+1];
    }
    n = n-1;
    display();
}
```

### Sample Output 1

-----MENU-----

1. CREATE
2. DISPLAY
3. INSERT
4. DELETE
5. EXIT

-----

ENTER YOUR CHOICE: 1

Enter the size of the array elements: 3

Enter the elements for the array: 10 25 30

ENTER YOUR CHOICE: 2

The array elements are: 10 25 30

ENTER YOUR CHOICE: 3

Enter the position for the new element: 1

Enter the element to be inserted : 20

ENTER YOUR CHOICE: 2

The array elements are: 10 20 25 30

ENTER YOUR CHOICE: 4

Enter the position of the element to be deleted: 3

The deleted element is =30

enter your choice: 5

Exit

### Sample Output 2

-----MENU-----

1. CREATE
2. DISPLAY
3. INSERT
4. DELETE
5. EXIT

-----

ENTER YOUR CHOICE: 1

Enter the size of the array elements: 3

Enter the elements for the array: 20 20 20

ENTER YOUR CHOICE: 2

The array elements are: 20 20 20

ENTER YOUR CHOICE: 3

Enter the position for the new element: 1

Enter the element to be inserted : 1

ENTER YOUR CHOICE: 2

The array elements are: 20 10 20 20

ENTER YOUR CHOICE: 4

Enter the position of the element to be deleted: 3

The deleted element is =20

enter your choice: 5

Exit

## EXPERIMENT-2

**Program 2: Design, Develop and Implement a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX)**

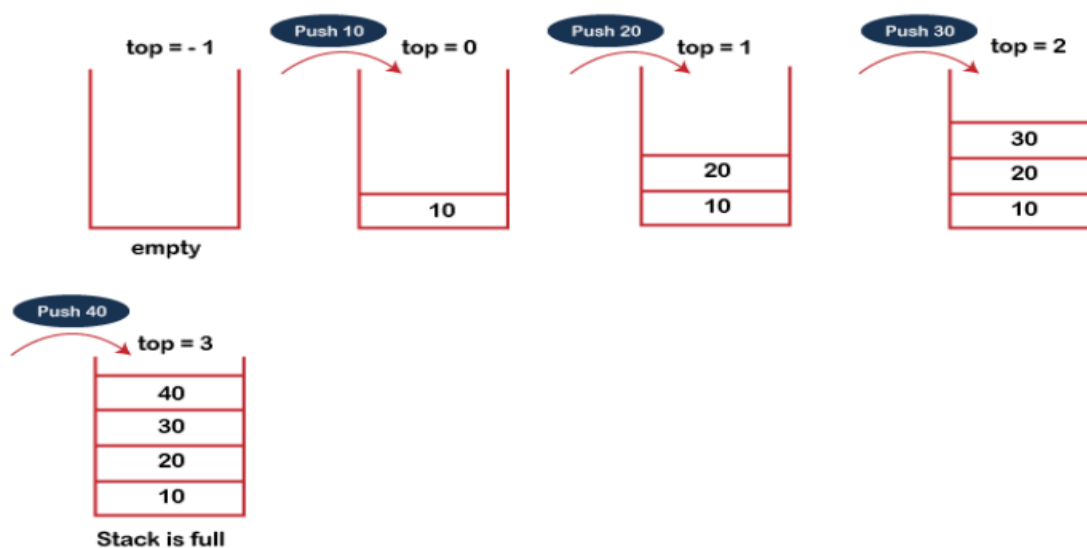
- Push*** an Element on to Stack
- Pop*** an Element from Stack
- Demonstrate** how Stack can be used to check *Palindrome*
- Demonstrate** *Overflow* and *Underflow* situations on Stack
- Display** the status of Stack
- Exit**

**Support the program with appropriate functions for each of the above operations**

### **PUSH operation: -**

The steps involved in the PUSH operation is given below:

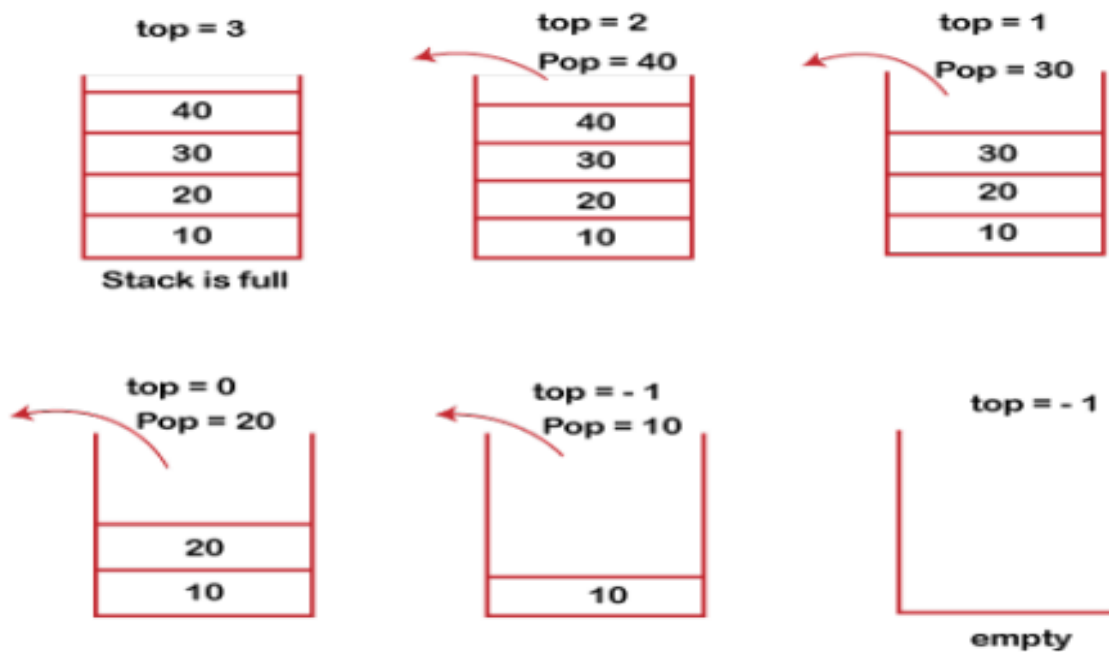
- Before inserting an element in a stack, we check whether the stack is full.
- If we try to insert the element in a stack, and the stack is full, then the overflow condition occurs.
- When we initialize a stack, we set the value of top as -1 to check that the stack is empty.
- When the new element is pushed in a stack, first, the value of the top gets incremented, i.e.,  $top = top + 1$ , and the element will be placed at the new position of the top.
- The elements will be inserted until we reach the max size of the stack.



### POP operation:-

The steps involved in the POP operation is given below:

- Before deleting the element from the stack, we check whether the stack is empty.
- If we try to delete the element from the empty stack, then the underflow condition occurs.
- If the stack is not empty, we first access the element which is pointed by the top. Once the pop operation is performed, the top is decremented by 1, i.e.,  $\text{top} = \text{top} - 1$ .



### ALGORITHM:

Step 1: Start.

Step 2: Initialize stack size MAX and top of stack -1.

Step 3: Push integer element on to stack and display the contents of the stack. if stack is full give a message as „Stack is Overflow“.

Step 4: Pop element from stack along with display the stack contents. if stack is empty give a message as „Stack is Underflow“.

Step 5: Check whether the stack contents are Palindrome or not.

Step 6: Stop.

### Program code:

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 5

int s[MAX];
int top = -1;

void push(int item);
int pop();
void palindrome();
void display();

void main()
{
    int choice, item;
    while(1)
    {
        printf("\n\n\n~~~~~Menu~~~~~ : ");
        printf("\n=>1.Push an Element to Stack and Overflow demo ");
        printf("\n=>2.Pop an Element from Stack and Underflow demo");
        printf("\n=>3.Palindrome demo ");
        printf("\n=>4.Display ");
        printf("\n=>5.Exit");
        printf("\n Enter your choice: ");
        scanf("%d", &choice);
        switch(choice)
        {
            case 1:      printf("\n Enter an element to be pushed: ");
                        scanf("%d", &item);
                        push(item);
```



```
                break;
        case 2:    item = pop();
                   if(item != -1)
                       printf("\n Element popped is: %d", item);
                   break;
        case 3:    palindrome();
                   break;
        case 4:    display();
                   break;
        case 5:    exit(1);
        default:   printf("\n Please enter valid choice ") ;
                   break;
    }
}
```

```
void push(int item)
{
    if(top == MAX-1)
    {
        printf("\n~~~Stack overflow~~~");
        return;
    }

    top = top + 1 ;
    s[top] = item;
}
```

```
int pop()
{
    int item;
```

```
if(top == -1)
{
    printf("\n~~~Stack underflow~~~");
    return -1;
}
item = s[top];
top = top - 1;
return item;
}
```

```
void display()
{
    int i;
    if(top == -1)
    {
        printf("\n~~~Stack is empty~~~");
        return;
    }
    printf("\n Stack elements are:\n ");
    for(i=top; i>=0 ; i--)
        printf("| %d \n", s[i]);
}
```

```
void palindrome()
{
    int flag=1,i;
    printf("\n Stack content are:\n");
    for(i=top; i>=0 ; i--)
        printf("| %d \n", s[i]);
    printf("\n Reverse of stack content are:\n");
    for(i=0; i<=top; i++)
```

```
printf("| %d \n", s[i]);

for(i=0; i<=top/2; i++)
{
    if( s[i] != s[top-i] )
    {
        flag = 0;
        break;
    }
}
if(flag == 1)
{
    printf("\n It is palindrome number");
}
else
{
    printf("\n It is not a palindrome number");
}
}
```

### **Output:**

```
~~~~~Menu~~~~~ :
=>1.Push an Element to Stack and Overflow demo
=>2.Pop an Element from Stack and Underflow demo
=>3.Palindrome demo
=>4.Display
=>5.Exit
```

Enter your choice: **1**

**Enter an element to be pushed: 11**

```
~~~~~Menu~~~~~ :
=>1.Push an Element to Stack and Overflow demo
=>2.Pop an Element from Stack and Underflow demo
=>3.Palindrome demo
=>4.Display
=>5.Exit
```

Enter your choice: 1

**Enter an element to be pushed: 12**

~~~~~Menu~~~~~ :

- =>1.Push an Element to Stack and Overflow demo
- =>2.Pop an Element from Stack and Underflow demo
- =>3.Palindrome demo
- =>4.Display
- =>5.Exit

Enter your choice: 1

**Enter an element to be pushed: 13**

~~~~~Menu~~~~~ :

- =>1.Push an Element to Stack and Overflow demo
- =>2.Pop an Element from Stack and Underflow demo
- =>3.Palindrome demo
- =>4.Display
- =>5.Exit

Enter your choice: 1

**Enter an element to be pushed: 14**

~~~~~Menu~~~~~

- =>1.Push an Element to Stack and Overflow demo
- =>2.Pop an Element from Stack and Underflow demo
- =>3.Palindrome demo
- =>4.Display
- =>5.Exit

Enter your choice: 1

**Enter an element to be pushed: 15**

~~~~~Menu~~~~~

- =>1.Push an Element to Stack and Overflow demo
- =>2.Pop an Element from Stack and Underflow demo
- =>3.Palindrome demo
- =>4.Display
- =>5.Exit

Enter your choice: 1

**Enter an element to be pushed: 16**

~~~~~Stack overflow~~~~~

~~~~~Menu~~~~~

- =>1.Push an Element to Stack and Overflow demo
- =>2.Pop an Element from Stack and Underflow demo
- =>3.Palindrome demo
- =>4.Display
- =>5.Exit

Enter your choice: 4

**Stack elements are:**

| 15 |  
| 14 |  
| 13 |  
| 12 |  
| 11 |

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo  
=>2.Pop an Element from Stack and Underflow demo  
=>3.Palindrome demo  
=>4.Display  
=>5.Exit

Enter your choice: 2

**Element popped is: 15**

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo  
=>2.Pop an Element from Stack and Underflow demo  
=>3.Palindrome demo  
=>4.Display  
=>5.Exit

Enter your choice: 4

**Stack elements are:**

| 14 |  
| 13 |  
| 12 |  
| 11 |

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo  
=>2.Pop an Element from Stack and Underflow demo  
=>3.Palindrome demo  
=>4.Display  
=>5.Exit

Enter your choice: 2

**Element popped is: 14**

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo  
=>2.Pop an Element from Stack and Underflow demo  
=>3.Palindrome demo  
=>4.Display  
=>5.Exit

Enter your choice: 2

**Element popped is: 13**

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo

=>2.Pop an Element from Stack and Underflow demo

=>3.Palindrome demo

=>4.Display

=>5.Exit

Enter your choice: 2

**Element popped is: 12**

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo

=>2.Pop an Element from Stack and Underflow demo

=>3.Palindrome demo

=>4.Display

=>5.Exit

Enter your choice: 2

**Element popped is: 11**

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo

=>2.Pop an Element from Stack and Underflow demo

=>3.Palindrome demo

=>4.Display

=>5.Exit

Enter your choice: 2

~~~~Stack underflow~~~~

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo

=>2.Pop an Element from Stack and Underflow demo

=>3.Palindrome demo

=>4.Display

=>5.Exit

Enter your choice: 4

~~~~Stack is empty~~~~

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo

=>2.Pop an Element from Stack and Underflow demo

=>3.Palindrome demo

=>4.Display

=>5.Exit

Enter your choice: 1

**Enter an element to be pushed: 11**

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo

=>2.Pop an Element from Stack and Underflow demo

=>3.Palindrome demo

=>4.Display

=>5.Exit

Enter your choice: **1**

**Enter an element to be pushed: 22**

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo

=>2.Pop an Element from Stack and Underflow demo

=>3.Palindrome demo

=>4.Display

=>5.Exit

Enter your choice: **1**

**Enter an element to be pushed: 11**

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo

=>2.Pop an Element from Stack and Underflow demo

=>3.Palindrome demo

=>4.Display

=>5.Exit

Enter your choice: **3**

**Stack content are:**

| **11** |

| **22** |

| **11** |

**Reverse of stack content are:**

| **11** |

| **22** |

| **11** |

**It is palindrome number**

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo

=>2.Pop an Element from Stack and Underflow demo

=>3.Palindrome demo

=>4.Display

=>5.Exit

Enter your choice: **2**

**Element popped is: 11**

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo

=>2.Pop an Element from Stack and Underflow demo

=>3.Palindrome demo

=>4.Display

=>5.Exit

Enter your choice: **2**

**Element popped is: 22**

~~~~~Menu~~~~~

- =>1.Push an Element to Stack and Overflow demo
- =>2.Pop an Element from Stack and Underflow demo
- =>3.Palindrome demo
- =>4.Display
- =>5.Exit

Enter your choice: 2

**Element popped is: 11**

~~~~~Menu~~~~~

- =>1.Push an Element to Stack and Overflow demo
- =>2.Pop an Element from Stack and Underflow demo
- =>3.Palindrome demo
- =>4.Display
- =>5.Exit

Enter your choice: 1

**Enter an element to be pushed: 11**

~~~~~Menu~~~~~

- =>1.Push an Element to Stack and Overflow demo
- =>2.Pop an Element from Stack and Underflow demo
- =>3.Palindrome demo
- =>4.Display
- =>5.Exit

Enter your choice: 1

**Enter an element to be pushed: 22**

~~~~~Menu~~~~~

- =>1.Push an Element to Stack and Overflow demo
- =>2.Pop an Element from Stack and Underflow demo
- =>3.Palindrome demo
- =>4.Display
- =>5.Exit

Enter your choice: 1

**Enter an element to be pushed: 33**

~~~~~Menu~~~~~

- =>1.Push an Element to Stack and Overflow demo
- =>2.Pop an Element from Stack and Underflow demo
- =>3.Palindrome demo
- =>4.Display
- =>5.Exit

Enter your choice: 3

**Stack content are:**

| 33 |

| 22 |



| 11 |

**Reverse of stack content are:**

| 11 |

| 22 |

| 33 |

**It is not a palindrome number**

~~~~~Menu~~~~~

=>1.Push an Element to Stack and Overflow demo

=>2.Pop an Element from Stack and Underflow demo

=>3.Palindrome demo

=>4.Display

=>5.Exit

Enter your choice: **5**