SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)
DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE CODE: DJS22ITL302 **DATE:19/10/2023**
COURSE NAME: Data Structure Laboratory **CLASS: I1-Batch1**

NAME: Ayush Vinod Upadhyay
ROLL NO: I025
SAP ID: 60003220131
BRANCH: Information Technology
BATCH: 1

EXPERIMENT NO. 01

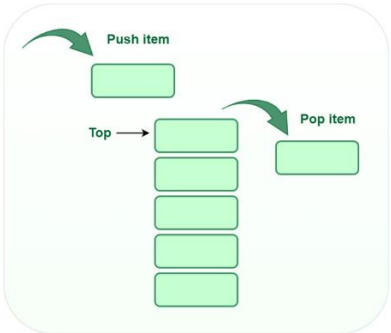
CO/LO: CO1

AIM: Implementation of Stack and Queues using Arrays.

THEORY:

A **Stack** is a linear data structure that follows a particular order in which the operations are performed. The order may be LIFO (Last In First Out) or FILO (First In Last Out). LIFO implies that the element that is inserted last, comes out first and FILO implies that the element that is inserted first, comes out last. The basic operations associated with a stack are:

- **Push**: Add an element to the top of the stack.
- **Pop**: Remove an element from the top of the stack.
- **Peek/Top**: Get the value of the top of the stack without removing it.
- **isEmpty**: Check if the stack is empty.



A **Queue** is a linear data structure that is open at both ends and the operations are performed in First In First Out (FIFO) order. The basic operations associated with a queue are:

- **Enqueue**: Add an element to the end of the queue.
- **Dequeue**: Remove an element from the front of the queue.
- **IsEmpty**: Check if the queue is empty.
- **IsFull**: Check if the queue is full.
- **Peek/Front**: Get the value of the front of the queue without removing it.

In terms of diagrams, a stack can be visualized as a vertical stack of items, where items are added or removed from the top. A queue can be visualized as a horizontal line (like a queue in a supermarket), where items are added at one end (the rear) and removed from the other end (the front).

COURSE CODE: DJS22ITL302 **DATE:19/10/2023**
COURSE NAME: Data Structure Laboratory **CLASS: I1-Batch1**

Queue Data Structure

STACK PROGRAM:

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

int top=-1, arr[SIZE];

void push(){
    int data;
    if (top==SIZE -1)
    {
        printf("Stack Overflow \n \n");
    }
    else
    {
        printf("Enter element to be inserted ");
        scanf("%d",&data);
        top=top+1;
        arr[top]=data;
        printf("%d is inserted\n \n", data);
    }
}
void pop(){
    if(top ==-1){
        printf("Stack Underflow \n \n");
    }else{
        printf("%d is deleted\n \n", arr[top]);
        top=top-1;
    }
}
void peek(){
```

COURSE CODE: DJS22ITL302 **DATE:19/10/2023**
COURSE NAME: Data Structure Laboratory **CLASS: I1-Batch1**

```
    if (top == -1)
    {
        printf("Underflow!!\n \n");
    }
    else
    {
        printf("Elements present in the stack: ");
        for (int i = top; i>=0 ; i--)
        {
            printf("%d ", arr[i]);
        }
        printf("\n \n");
    }
}
int  main()
{
    int choice=-1;
    while (choice != 0)
    {
        printf("Enter 1 to push \nEnter 2 to pop \nEnter 3 to peek \nEnter 0
to exit \n");
        scanf("%d", &choice);
        if (choice==1)
        {
            push();
        }else if (choice==2)
        {
            pop();
        }else if (choice==3)
        {
            peek();
        }else if (choice==0)
        {
            break;
        }
    }
return 0;
}
```

STACK EXECUTION:

COURSE CODE: DJS22ITL302 **DATE:19/10/2023**
COURSE NAME: Data Structure Laboratory **CLASS: I1-Batch1**

```
Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
3
Elements present in the stack: 20 10

Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
1
Enter element to be inserted 30
30 is inserted

Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
3
Elements present in the stack: 30 20 10
```

```
Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
3
Elements present in the stack: 30 20 10

Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
2
30 is deleted

Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
3
Elements present in the stack: 20 10
```

QUEUE PROGRAM:

```c
#include<stdio.h>
#include<stdlib.h>
#define SIZE 10

int front=-1, rear=-1, arr[SIZE];

void insert(){
    int data;
    printf("Enter element to be inserted ");
    scanf("%d",&data);
    if (rear==SIZE-1)
    {
        printf("Stack Overflow\n \n");
    }
    else if (front==-1 && rear==-1)
    {
        front=rear=0;
        arr[rear]=data;
        printf("%d is inserted\n \n",data);
    }
    else{
```

COURSE CODE: DJS22ITL302 **DATE:19/10/2023**
COURSE NAME: Data Structure Laboratory **CLASS: I1-Batch1**

```
        rear++;
        arr[rear]=data;
        printf("%d is inserted\n \n",data);
    }
}
void delete(){
    if(front == -1){
        printf("\nUnderflow \n \n");
    }else{
        printf("%d is deleted from stack\n \n",arr[front]);
        front++;
    }
}
void traverse(){
    for (int i = front; i <= rear; i++)
    {
        printf("%d ",arr[i]);
    }
    printf("\n \n");
}

int  main()
{
    int choice=-1;
    while (choice != 0)
    {
        printf("Enter 1 to push \nEnter 2 to pop \nEnter 3 to peek \nEnter 0
to exit \n");
        scanf("%d", &choice);
        if (choice==1)
        {
            insert();
        }else if (choice==2)
        {
            delete();
        }else if (choice==3)
        {
            traverse();
        }else if (choice==0)
        {
            break;
        }
```

COURSE CODE: DJS22ITL302 **DATE:19/10/2023**
COURSE NAME: Data Structure Laboratory **CLASS: I1-Batch1**

```
    }
return 0;
}
```

QUEUE EXECUTION:

```
Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
3
10 20

Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
1
Enter element to be inserted 30
30 is inserted

Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
3
10 20 30
```

```
Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
3
10 20 30

Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
2
10 is deleted from stack

Enter 1 to push
Enter 2 to pop
Enter 3 to peek
Enter 0 to exit
3
20 30
```

CONCLUSION: I learnt implementation of Stack and Queue using Array and performed Insertion and Deletion operations on them.

REFERENCE: GeeksForGeeks , W3School for Theory

Self Implemented the code