Programmer Name: Param Chordiya
Batch: E5
Problem Statement:

1. Implement stack and queue using struct and perform push, pop operations on it.
2. Implement queue using arrays and perform insert/enque, delete/deque operations on it.

1) Stack using array.

Input:

```
#include <stdio.h>
#define MAXSIZE 100

struct stack
{
   int stk[MAXSIZE];
   int top;
};
typedef struct stack STACK;
STACK s;

void push(void);
void pop(void);
void display(void);

void main ()
{
                    printf("\n*************************************\n\t        ROLL
NO:22119\n*************************************\n");
    int choice;
    int option = 1;
    s.top = -1;

    printf ("STACK OPERATION\n");
    while (option)
```

```
    {
       printf ("----------------------------------------\n");
       printf ("1)PUSH\n");
       printf ("2)POP\n");
       printf ("3)DISPLAY\n");
       printf ("4)EXIT\n");
       printf ("----------------------------------------\n");

       printf ("Enter your choice\n");
       scanf  ("%d", &choice);
       switch (choice)
       {
       case 1:
          push();
          break;
       case 2:
          pop();
          break;
       case 3:
          display();
          break;
       case 4:
          return;
       }
       fflush (stdin);
       printf ("Do you want to continue(Type 0 or 1)?\n");
       scanf    ("%d", &option);
    }
}

void push ()
{
   int num;
   if (s.top == (MAXSIZE - 1))
   {
      printf ("Stack is Full\n");
      return;
```

```
    }
    else
    {
        printf ("Enter the element to be pushed\n");
        scanf ("%d", &num);
        s.top = s.top + 1;
        s.stk[s.top] = num;
    }
    return;
}

void pop ()
{
    int num;
    if (s.top == - 1)
    {
        printf ("Stack is Empty\n");
        return (s.top);
    }
    else
    {
        num = s.stk[s.top];
        printf ("poped element is = %dn", s.stk[s.top]);
        s.top = s.top - 1;
    }
    return(num);
}

void display ()
{
    int i;
    if (s.top == -1)
    {
        printf ("Stack is empty\n");
        return;
    }
    else
```

```
    {
        printf ("\n The status of the stack is \n");
        for (i = s.top; i >= 0; i--)
        {
            printf ("%d\n", s.stk[i]);
        }
    }
    printf ("\n");
}
```

Output:

```
**********************************
          ROLL NO:22119
**********************************
STACK OPERATION
----------------------------------------
1)PUSH
2)POP
3)DISPLAY
4)EXIT
----------------------------------------
Enter your choice
1
Enter the element to be pushed
5
Do you want to continue(Type 0 or 1)?
1
----------------------------------------
1)PUSH
2)POP
3)DISPLAY
4)EXIT
----------------------------------------
Enter your choice
1
Enter the element to be pushed
9
Do you want to continue(Type 0 or 1)?
1
----------------------------------------
1)PUSH
2)POP
3)DISPLAY
4)EXIT
----------------------------------------
Enter your choice
1
Enter the element to be pushed
7
Do you want to continue(Type 0 or 1)?
1
```

```
-----------------------------------------
1)PUSH
2)POP
3)DISPLAY
4)EXIT
-----------------------------------------
Enter your choice
3

 The status of the stack is
7
9
5

Do you want to continue(Type 0 or 1)?
1
-----------------------------------------
1)PUSH
2)POP
3)DISPLAY
4)EXIT
-----------------------------------------
Enter your choice
2
poped element is = 7
Do you want to continue(Type 0 or 1)?
1
-----------------------------------------
1)PUSH
2)POP
3)DISPLAY
4)EXIT
-----------------------------------------
Enter your choice
3

 The status of the stack is
9
5

Do you want to continue(Type 0 or 1)?
```

2) Queue using array:

INPUT:

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

struct queue
{
                    int arr[max];
                    int front;
                    int rear;
} s;

void enque(struct queue *s);
void deque(struct queue *s);
void display(struct queue *s);

void main()
{
                    printf("\n*********************************************\n\t
QUEUE USING STRUCT");
                    printf("\n*********************************************\n\t\t
ROLL NO:22119\n*********************************************\n");
                    s.front = -1;
                    s.rear = -1;
                    int c,choice;

                    do
                    {
                        printf("1.Insert/EnQueue\n2.Delete/DeQueue\n3.Display\n");

                    printf("==========================================
");
                        printf("\nEnter your choice:");
                        scanf("%d",&c);
                        switch(c)
```

```
        {
            case 1:
                    enque(&s);
                    break;
            case 2:
                    deque(&s);
                    break;
            case 3:
                    display(&s);
                    break;
            default:
                    printf("\nEnter valid choice.");
        }
             printf("\nPress 1 to continue and 0 to stop. \n");
             scanf("%d",&choice);
        }
            while(choice==1);
            getch();
}

void enque(struct queue *s)
{
                    int x;
                    if(s->rear == (max-1))
                    {
                        printf("Overflow");
                    }
                    else if(s->front == -1 && s->rear == -1)
                    {
                        s->front = s->rear = 0;
                        printf("\nEnter the value which you want to insert:");
                        scanf("%d",&x);
                        s->arr[s->rear]=x;
                    }
                    else
                    {
                        s->rear++;
```

```
            printf("\nEnter the value which you want to insert : ");
            scanf("%d",&x);
            s->arr[s->rear]=x;
        }
}

void deque(struct queue *s)
{
            if(s->front == -1 && s->rear == -1)
            {
                printf("Underflow");
            }
            else if(s->front == s->rear)
            {
                s->front = s->rear = -1;
            }
            else
            {
                s->front++;
            }
}
void display(struct queue *s)
{
            int i;
            if(s->front == -1 && s->rear == -1)
            {
                printf("Underflow");
            }
            else
            {
                printf("The elements in the queue are :\n");
                for(i=s->front;i<s->rear+1;i++)
                {
                        printf("%d\n",s->arr[i]);
                }
            }
}
```

OUTPUT:

```
*****************************************
            QUEUE USING STRUCT
*****************************************
              ROLL NO:22119
*****************************************
1.Insert/EnQueue
2.Delete/DeQueue
3.Display
=========================================
Enter your choice:1

Enter the value which you want to insert:5

Press 1 to continue and 0 to stop.
1
1.Insert/EnQueue
2.Delete/DeQueue
3.Display
=========================================
Enter your choice:1

Enter the value which you want to insert : 8

Press 1 to continue and 0 to stop.
1
1.Insert/EnQueue
2.Delete/DeQueue
3.Display
=========================================
Enter your choice:1

Enter the value which you want to insert : 9

Press 1 to continue and 0 to stop.
1
1.Insert/EnQueue
2.Delete/DeQueue
3.Display
=========================================
Enter your choice:3
The elements in the queue are :
5
8
9
```

```
Press 1 to continue and 0 to stop.
1
1.Insert/EnQueue
2.Delete/DeQueue
3.Display
==========================================
Enter your choice:2

Press 1 to continue and 0 to stop.
1
1.Insert/EnQueue
2.Delete/DeQueue
3.Display
==========================================
Enter your choice:3
The elements in the queue are :
8
9

Press 1 to continue and 0 to stop.
```