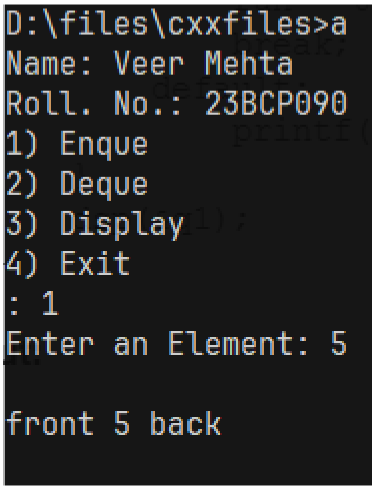
Data Structures Lab Assignment 6
Name: Veer Mehta
Roll. No.: 23BCP090

Q1. Write an implementation of Circular Queue using Arrays
Code:

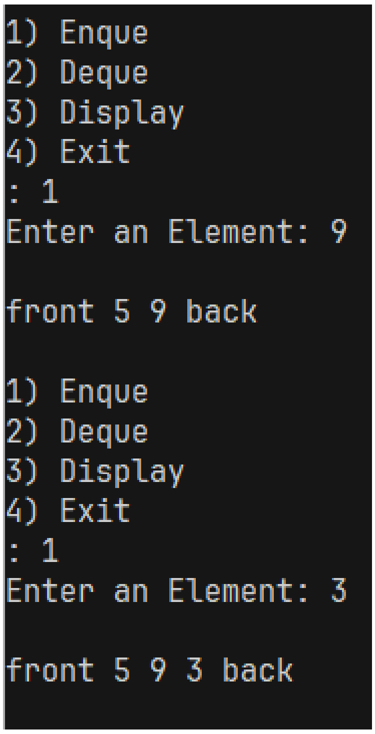
```
#include <stdio.h>
#include <stdlib.h>
int mx = 10;
struct queue
{
    int *que;   //queue array
    int f;      //front
    int b;      //back
    int sz;
};
int isuf(struct queue *q)   //is underflow
{
    return !(q->sz);
}
int isof(struct queue *q)   //is overflow
{
    return q->sz == mx;
}
int dq(struct queue *q)
{
    int x = q->que[q->f];
    q->f = (q->f+1)%mx;
    q->sz--;
    return x;
}
void nq(struct queue *q, int x)
{
    q->que[q->b] = x;
    q->b = (q->b+1)%mx;
    q->sz++;
}
void dsp(struct queue *q)
{
    if (isuf(q))
    {
        printf("UNDERFLOW\n");
        return;
    }
    printf("\nfront ");
    for (int i = 0; i < q->sz; i++)
        printf("%d ", q->que[(q->f+i)%mx]);
    printf("back\n\n");
}
```

```
void main()
{
    printf("Name: Veer Mehta\nRoll. No.: 23BCP090\n");
    struct queue q1 = {(int*)malloc(sizeof(int[mx])), 0, 0, 0};
    int x, c, run = 1;

    while (run) {
        printf("1) Enque\n2) Deque\n3) Display\n4) Exit\n: ");
        scanf("%d", &c);
        switch (c)
        {
            case 1:
                if (isof(&q1))
                    printf("OVERFLOW\n");
                else
                {
                    printf("Enter an Element: ");
                    scanf("%d", &x);
                    nq(&q1, x);
                }
                break;
            case 2:
                if (isuf(&q1))
                    printf("UNDERFLOW\n");
                else
                    printf("The Dequed element: %d\n", dq(&q1));
                break;
            case 3:
                break;
            case 4:
                run = 0;
                break;
            default:
                printf("Enter a valid Command\n");
        }
        dsp(&q1);
    }
}
```

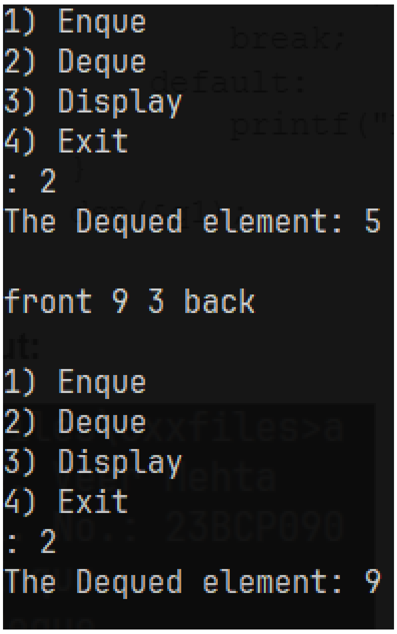
Output:



```
D:\files\cxxfiles>a
Name: Veer Mehta
Roll. No.: 23BCP090
1) Enque
2) Deque
3) Display
4) Exit
: 1
Enter an Element: 5

front 5 back
```



```
1) Enque
2) Deque
3) Display
4) Exit
: 1
Enter an Element: 9

front 5 9 back

1) Enque
2) Deque
3) Display
4) Exit
: 1
Enter an Element: 3

front 5 9 3 back
```



```
1) Enque
2) Deque
3) Display
4) Exit
: 2
The Dequed element: 5

front 9 3 back

1) Enque
2) Deque
3) Display
4) Exit
: 2
The Dequed element: 9
```