

Experiment 3: Write C programs to implement a double ended queue ADT using

i) array

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  struct node
4  {
5      int data;
6      struct node *next;
7  };
8  struct node *front;
9  struct node *rear;
10 void insert();
11 void delete();
12 void display();
13 void main ()
14 {
15     int choice;
16     while(choice != 4)
17     {
18         printf("\n*****Main Menu*****\n");
19         printf("\n=====");
20         printf("\n1.insert an element\n2.Delete an element\n3.Display the queue\n4.Exit\n");
21         printf("\nEnter your choice ?");
22         scanf("%d",& choice);
23
24         switch(choice)
25         {
26             case 1:
27                 insert();
28                 break;
29             case 2:
30                 delete();
31                 break;
32             case 3:
33                 display();
34                 break;
35             case 4:
36                 exit(0);
37                 break;
38             default:
39                 printf("\nEnter valid choice??\n");
40         }
41     }
42
43     void insert()
44     {
45         struct node *ptr;
```

```

46
47 ptr = (struct node *) malloc (sizeof(struct node));
48 if(ptr == NULL)
49 {
50 printf("\nOVERFLOW\n");
51 return;
52 }
53 else
54 {
55 printf("\nEnter value?\n");
56 scanf("%d",&item);
57 ptr -> data = item;
58 if(front == NULL)
59 {
60 front = ptr;
61 rear = ptr;
62 front -> next = NULL;
63 rear -> next = NULL;
64 }
65 else
66 {
67 rear -> next = ptr;
68 rear = ptr;
69 rear->next = NULL;
70 }
71 }
72 }
73 void delete ()
74 {
75 struct node *ptr;
76 if(front == NULL)
77 {
78 printf("\nUNDERFLOW\n");
79 return;
80 }
81 else
82 {
83 ptr = front;
84 front = front -> next;
85 free(ptr);
86 }
87 }
88 void display()
89 {
90 struct node *ptr;
91 ptr = front;
92 if(front == NULL)
93 {
94 printf("\nEmpty queue\n");
95 }
96 else
97 { printf("\nprinting values ..... \n");
98 while(ptr != NULL)
99 {
100 printf("\n%d\n",ptr -> data);
101 ptr = ptr -> next;
102 }
103 }
104 }

```

Output:

```
D:\Academics\3rd Sem\DSUC x + v

*****Main Menu*****

=====

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice ?1
Enter value?
1

*****Main Menu*****

=====

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice ?1
Enter value?
1
```

```
D:\Academics\3rd Sem\DSUC x + v

1

*****Main Menu*****

=====

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice ?1
Enter value?
2

*****Main Menu*****

=====

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice ?1
Enter value?
3
```

```
D:\Academics\3rd Sem\DSUC  X + v

*****Main Menu*****

=====

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice ?3

printing values .....

1
1
2
3

*****Main Menu*****

=====

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice ?2
```

```
D:\Academics\3rd Sem\DSUC  X + v

*****Main Menu*****

=====

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice ?3

printing values .....

1
2
3

*****Main Menu*****

=====

1.insert an element
2.Delete an element
3.Display the queue
4.Exit

Enter your choice ?4

-----

Process exited after 24.3 seconds with return value 0
Press any key to continue . . . |
```

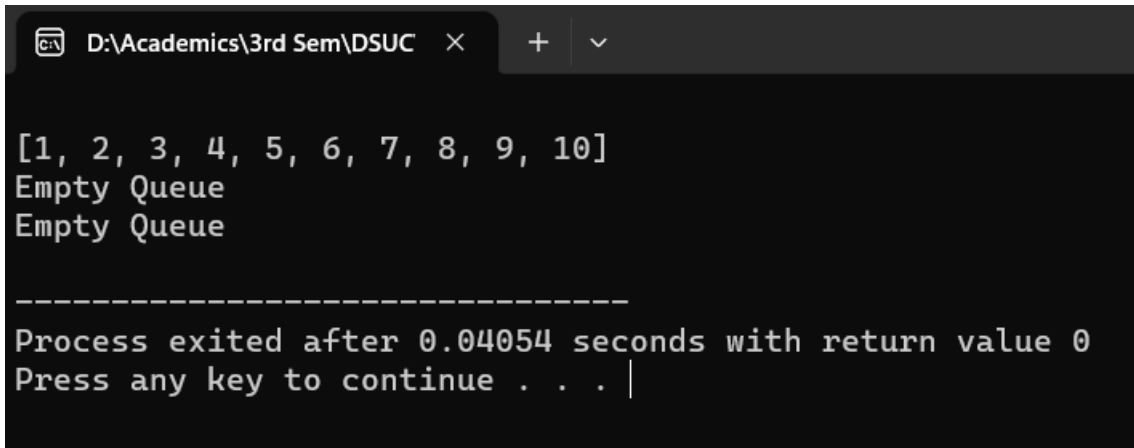
ii) doubly linked list respectively.

LinkedListImplementationArray.c

```
1  #include <stdio.h>
2  #define MAX 10
3  int front = 0, rear = 0, i;
4  int queue[MAX];
5  int isFull() {
6      return MAX == rear;
7  }
8  int isEmpty() {
9      return front == rear;
10 }
11 void enqueue(int data) {
12     if (isFull()) {
13         printf("Overflow");
14         return;
15     }
16     queue[rear] = data;
17     rear++;
18 }
19 void dequeue() {
20     if (isEmpty()) {
21         printf("Empty Queue\n");
22         return;
23     }
24     front++;
25 }
26 int peek() {
27     if (isEmpty()) {
28         printf("Empty Queue\n");
29         return -999;
30     }
31     return queue[front];
32 }
33 void printQueue() {
34     if (isEmpty()) {
35         printf("Empty Queue\n");
36         return;
37     }
38     printf("\n[");
39     for (i = front; i < rear; i++) {
40         if (i == rear - 1) {
41             printf("%d", queue[i]);
42         } else {
43             printf("%d, ", queue[i]);
44         }
45     }
46     printf("]\n");
47 }
48
49 int main() {
50     for (i = 1; i <= 10; i++) enqueue(i);
51     printQueue();
```

```
52   for (i = 1; i <= 11; i++) dequeue();
53   printQueue();
54   return 0;
55 }
```

Output:



The screenshot shows a terminal window with the title bar "D:\Academics\3rd Sem\DSUC". The output of the program is as follows:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Empty Queue
Empty Queue

-----
Process exited after 0.04054 seconds with return value 0
Press any key to continue . . . |
```