WEEK-4

1) Queue using Array

```
#include <stdio.h>
#include <stdlib.h>
#define que_size 5
int item, front=0, rear=-1, q[10];
void insertrear()
{
    if (rear == que_size -1)
    {
        printf("\n queue is full //overflow\n");
        return;
    }
    printf("\n Enter item to be inserted\n");
    scanf("%d", &item);
    rear = rear + 1;
    q[rear] = item;
}
intdeletefront()
{
    if (front > rear)
    {
        front=0;
        rear=-1;
        return -1;
    }
    return q[front++];
}
void display Q()
{
    int i;
```

```
if (front > rear)
{
    printf("\n Queue is empty \n");
    return;
}
printf("\n Contents of queue \n");
for (i=front; i<=rear; i++)
    printf("%d", q[i]);
}
int main()
{
    int choice;
    for(;;)
    {
        printf("\n 1. Insert rear \n 2. Delete front \n 3. Display \n 4. Exit \n");
        printf("Enter choice \n");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1: insertrear()
                    break;
            case 2: item = deletefront();
                    if (item == -1)
                     printf("Queue is Empty // Underflow \n");
                    else
                     printf("Item deleted = %d \n", item);
                    break;
            case 3: display Q();
                    break
            default : exit(0);
        }
    }
}
```

output

1. insert rear
2. Delete front
3. Display
4. Exit

Enter the choice

1

Enter element to be inserted

12

1. insert rear
2. delete front
3. Display
4. Exit.

Enter the choice

4.

2) Queue using Arrays

```
#include <stdio.h>
#include <stdlib.h>
#define que_size 5
int item, front=0, rear=-1, q[que_size], count=0;
void insertrear()
{
    if (count==que_size)
    {
        printf("\n Queue is full \\ overflow \n");
        return;
    }
```

```
printf ("\n Enter item to be inserted : ");
scanf ("%d", &item);
rear = (rear+1) % que_size;
q[rear] = item;
count++;
}
int deletefront()
{
if (count==0) return -1;
item = q[front];
front = (front+1) % que_size;
count--;
return item;
}
void display()
{
int i, f;
if (count==0)
{
printf ("\n Queue is empty \n");
return;
}
f=front;
printf ("\n Contents of queue \n");
for (i=0; i<count; i++)
{
printf ("%d\n", q[f]);
f = (f+1) % que_size;
}
}
void main()
{
int choice;
```

```
for(;;)
{
    printf("\n1. Insert rear \n2. Delete front \n3. Display \n
            4. Exit \n");
    printf("Enter choice: ");
    scanf("%d", &choice);
    switch(choice)
    {
        case 1 : insert_rear();
                 break;
        case 2 : item = delete_front();
                 if(item == -1)
                   printf("\n Queue is empty //Underflow \n");
                 else
                 printf("Item deleted is: %d \n", item);
                 break;
        case 3 : display();
                 break;
        default : exit(0);
    }
  }
}
```

Output:

1. insert rear
2. delete front
3. Display
4. Exit

Enter choice : 2.

Queue is empty // underflow

1. Insert rear
2. delete front
3. Display
4. Exit

Enter choice : 4.