

Practice : Linear Queue :

```
#include <stdio.h>
```

```
#define Que_Size 3
```

```
int item, front = 0, rear = -1, q[10];
```

```
void insertrear()
```

```
{ if (rear == Que_Size - 1)
```

```
{ printf("Queue overflow\n");  
return; }
```

```
rear = rear + 1
```

```
q[rear] = item; }
```

```
int deletefront()
```

```
{ if (front > rear)
```

```
{ front = 0; rear = -1;
```

```
return -1; }
```

```
return q[front++];
```

```
void displayQ()
```

```
{
```

```
    int i;
```

```
    if (front > rear)
```

```
{
```

```
        printf("Queue is empty\n");
```

```
        return;
```

```
}
```

```
    printf("Content of Queue\n");
```

```
    for (i = front; i <= rear; i++)
```

```
{
```

```
        printf("%d\n", q[i]);
```

```
}
```

```
}
```

```
void main()
```

```
{
```

```
    int choice;
```

```
    for (i = 1;
```

```
{
```

```
        printf("\n1 : insert rear
```

```
        \n2 : delete front
```

```
        \n3 : display
```

```
        \n4 : exit\n");
```

```
    printf("Enter the choice);
```

```
    scanf("%d", &choice);
```

```
    switch (choice)
```

```
{
```

```
        case 1 : printf("Enter the item to be inserted");
```

```
        scanf("%d", &item);
```

```
        insertrear();
```

```
        break;
```

```
        case 2 : item = deletefront();
```

```
        if (item == -1)
```

```
            printf("Queue is empty\n");
```

```
        else  
            printf("Item deleted = %d\n", item);
```



```
break;  
case 3: displayQ();  
        break;  
default: exit(0);  
}
```