

Queue

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

```
void enqueue(void);
void dequeue(void);
void show(void);
int int_arr[SIZE];
int Rear = -1;
int Front = -1;
```

```
int main(){
```

```
    int ch;
    while(1){
```

```
        printf("\n1.Enqueue\n2.Dequeue\n3.Display Queue\n4.Exit\nEnter your choice (1-4):");
        scanf("%d", &ch);
```

```
        switch(ch){
```

```
            case 1:
```

```
                enqueue();
                break;
```

```
            case 2:
```

```
                dequeue();
                break;
```

```
            case 3:
```

```
                show();
                break;
```

```
            case 4: exit(0);
```

```
            default:
```

```
                printf("Incorrect choice);
```

```
            }
```

```
        }
```

```
    }
```

Queue

EdTC-B3
20070123120
Parth Nikam

Camlin	Page
Date	/ /

```
void enqueue() {  
    int insert_item;  
    if (Rear == SIZE - 1)  
        printf("Overflow\n");  
    else {  
        if (Front == -1) {  
            Front = 0;  
            printf("Element to be inserted:");  
            scanf("%d", &insert_item);  
            Rear = Rear + 1;  
            inp_arr[Rear] = insert_item;  
        }  
    }  
}
```

```
void dequeue() {  
    if (Front == -1 || Front > Rear) {  
        printf("Underflow\n");  
        return;  
    }  
    else {  
        printf("Element to be deleted %d\n", inp_arr[Front]);  
        Front = Front + 1;  
    }  
}
```

```
void show() {  
    if (Front == -1)  
        printf("Empty Queue");  
    else {  
        printf("Queue\n");  
        for (int i = Front; i <= Rear; i++) {  
            printf("%d", inp_arr[i]);  
        }  
    }  
}
```