

# Code 1 : Circular Queue

```
// circular queue implementation
```

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

```
#define SIZE 100
```

```
int items[SIZE];
```

```
int front = -1, rear = -1;
```

```
int isFull(){
```

```
    if((front == rear + 1) || (front == 0 && rear == SIZE - 1)) return 1;
```

```
    return 0;
```

```
}
```

```
int isEmpty(){
```

```
    if(front == -1) return 1;
```

```
    return 0;
```

```
}
```

```
void enqueue (int element){
```

```
    if(isFull()){
```

```
        printf("cant enqueue, its full\n");
```

```
        return;
```

```
    }
```

```
    if(front == -1) front = 0;
```

```
    rear = (rear + 1) % SIZE;
```

```
    items[rear] = element;
```

```
    printf("pushed %d\n", element);
```

```
}
```



```
int dequeue () {  
    if (isEmpty()) {  
        printf("cant pop, its empty\n");  
        return;  
    }  
    int element = items[front];  
    if (front == rear) {  
        front = -1;  
        rear = -1;  
    }  
    else {  
        front = (front + 1) % SIZE;  
    }  
    printf("poped %d\n", element);  
    return (element);  
}
```

```
int main() {  
    enqueue(1);  
    enqueue(2);  
    enqueue(3);  
  
    dequeue();  
    dequeue();  
    return 0;  
}
```



C:\Users\bmsce\Desktop\1bm



# Output of Code 1

```
pushed 1  
pushed 2  
pushed 3  
poped 1  
poped 2
```

```
Process returned 0 (0x0)    execution time : 0.035 s  
Press any key to continue.
```





// singly linked list implementation

## Code 2: Singly Linked list

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

```
struct node{
    int value;
    struct node *next;
};
```

```
void displayLinkedList(struct node *p) {
    printf("Printing linked list\n");
    while( p != NULL) {
        printf("%d\n", p->value);
        p = p -> next;
    }
}
```

```
int main() {
    struct node *head;
    struct node *one = NULL;
    struct node *two = NULL;
    struct node *three = NULL;

    one = malloc(sizeof(struct node));
    two = malloc(sizeof(struct node));
    three = malloc(sizeof(struct node));

    one->value = 1;
    two->value = 2;
    three -> value = 3;
```



```
10
11 void displayLinkedList(struct node *p) {
12     printf("Printing linked list\n");
13     while( p != NULL) {
14         printf("%d\n", p-> value);
15         p = p -> next;
16     }
17 }
```

```
18
19 int main() {
20     struct node *head;
21     struct node *one = NULL;
22     struct node *two = NULL;
23     struct node *three = NULL;
24
25     one = malloc(sizeof(struct node));
26     two = malloc(sizeof(struct node));
27     three = malloc(sizeof(struct node));
28
29     one-> value = 1;
30     two-> value = 2;
31     three -> value = 3;
32
33     one -> next = two;
34     two -> next = three;
35     three -> next = NULL;
36
37     head = one;
38     displayLinkedList(head);
39 }
40
```



C:\Users\bmsce\Desktop\1bm X + v

## Output of code 2

Printing linked list

1  
2  
3

Process returned 0 (0x0) execution time : 0.038 s

Press any key to continue.