

10. Illustrate the concept of inter-process communication using message queue with a C program.

**Aim:**

To implement inter-process communication (IPC) using message queues in C.

**Algorithm:**

1. Create a message queue using `msgget()`.
2. Send a message to the queue using `msgsnd()`.
3. Receive the message from the queue using `msgrcv()`.
4. Display the received message.
5. Terminate the processes and clean up the resources.

**Procedure:**

1. Create a message queue with a unique key.
2. Define a structure for the message.
3. Use `msgsnd()` in the sender process to send a message to the queue.
4. Use `msgrcv()` in the receiver process to read the message from the queue.
5. Display the received message.
6. Clean up by removing the message queue when no longer needed.

**CODE:**

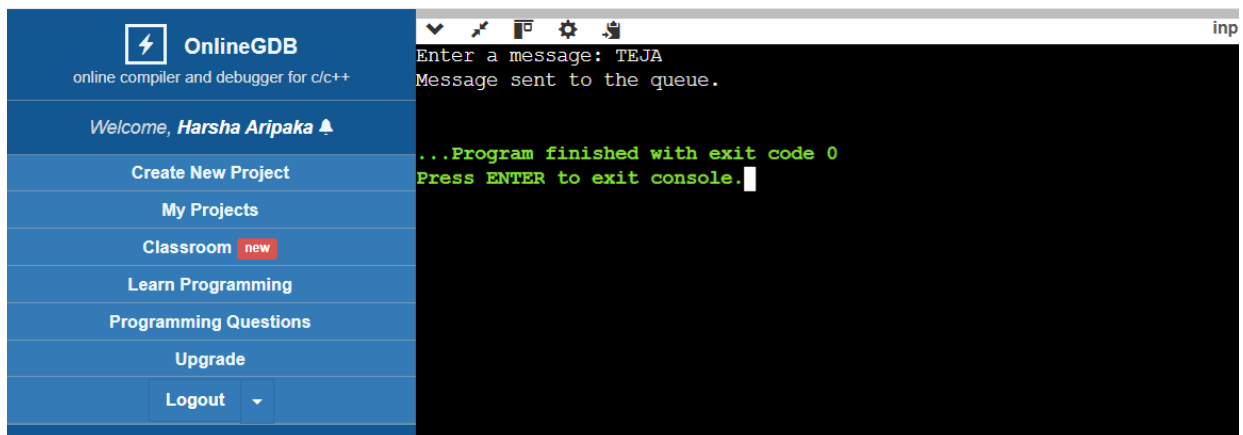
```
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <string.h>
```

```
#define MSG_SIZE 1024
```

```
struct msg_buffer {  
    long msg_type;  
    char msg_text[MSG_SIZE];  
};
```

```
int main() {  
    key_t key = 1234;  
    int msgid;  
    struct msg_buffer message;  
    msgid = msgget(key, 0666 | IPC_CREAT);  
    message.msg_type = 1;  
    printf("Enter a message: ");  
    fgets(message.msg_text, MSG_SIZE, stdin);  
    msgsnd(msgid, &message, sizeof(message), 0);  
    printf("Message sent to the queue.\n");  
    return 0;  
}
```

OUTPUT:

The screenshot shows the OnlineGDB web interface. On the left is a blue sidebar with navigation links: 'Create New Project', 'My Projects', 'Classroom' (with a 'new' badge), 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The main area on the right has a dark background and displays the program's execution output. The output text is: 'Enter a message: TEJA', 'Message sent to the queue.', and '...Program finished with exit code 0'. At the bottom of the output, it says 'Press ENTER to exit console.' with a cursor icon. The top of the main area has a toolbar with icons for file operations and a tab labeled 'inp'.

**Result:**

The C program successfully demonstrates inter-process communication using message queues. The sender process sends a message to the message queue, and the receiver process retrieves and displays the message from the queue.