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

Message Queue Program

```
// Common Header File: msg_def.h
#ifndef MSG_DEF_H
#define MSG_DEF_H

#define MAX 100
#define MSG_KEY 1234

struct msg_buffer {
   long msg_type;
   char msg_text[MAX];
};
#endif
```

Sender Program

```
// sender.c
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include <string.h>
#include "msg_def.h"

int main() {
   struct msg_buffer msg;
   int msgid = msgget(MSG_KEY, 0666 | IPC_CREAT);

   if (msgid == -1) {
```

```
        perror("msgget failed");
        return 1;
    }

    msg.msg_type = 1;  // Message type
    printf("Enter message to send: ");
    fgets(msg.msg_text, MAX, stdin);

    if (msgsnd(msgid, &msg, strlen(msg.msg_text) + 1, 0) == -1) {
        perror("msgsnd failed");
        return 1;
    }

    printf("Message sent: %s\n", msg.msg_text);
    return 0;
}
```

Receiver Program

```
// receiver.c
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/msg.h>
#include "msg_def.h"

int main() {
    struct msg_buffer msg;
    int msgid = msgget(MSG_KEY, 0666);

    if (msgid == -1) {
        perror("msgget failed");
```

```
    return 1;
  }

  if (msgrcv(msgid, &msg, sizeof(msg.msg_text), 1, 0) == -1) {
    perror("msgrcv failed");
    return 1;
  }

  printf("Message received: %s\n", msg.msg_text);

  // Optional: delete the message queue
  msgctl(msgid, IPC_RMID, NULL);

  return 0;
}
```

Sample Output

- Message Queue

  "Hello World I Am An AI Machine".

- Sender Program

  Message Sent: "Hello World I Am An AI Machine".

- Receiver Program

  Message received: "Hello World I Am An AI Machine".