```
/*
Computer Networks, Lab exam practice
Date:
*/

/* FILL THIS BEFORE PROCEEDING
Name:
Roll number:
IP address:
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <strings.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>

#define MAX_LINE 256

int main(int argc, char * argv[]){
  FILE *fp;
  struct hostent *hp;
  struct sockaddr_in sin;
  char *host;
  char buf[MAX_LINE];
  int s;
  int len;
  unsigned short SERVER_PORT;

   /*  Code to handle command line arguments.
       the first argument must be the remote IP address
       and the second argument must be the remote port number.

       if both the arguments are not present, print an error
       message and exit.
    */
  /*
  if(){


  }
  else {

  }
  */
  // END OF THE COMMAND LINE PARSING

  // For the prctice exam, you can start with the following...
  if (argc==3) {
    host = argv[1];
    SERVER_PORT = atoi(argv[2]);
  }
  else {
    fprintf(stderr, "usage: client host port\n");
    exit(1);
  }


  /* build address data structure */
```

```
    /* translate host name into peer's IP address */
    hp = gethostbyname(host);
    if (!hp) {
      /* print unknown host error message and exit */


    }

    /* Add code to build address data structure sin*/
    /* Add code to zero out the data structure memory */

    /* Set address family to AF_INET*/
    // = AF_INET;

    /* Set destination IP address */
    //bcopy(hp->h_addr, (char *)&sin.sin_addr, hp->h_length);

    /* Set destination port */
    // sin.sin_port = htons(server_port);

    /* The following fragment implements the above four lines*/
    /* It will not be given in the exam. You will have to write yourself. */
    bzero((char *)&sin, sizeof(sin));
    sin.sin_family = AF_INET;
    bcopy(hp->h_addr, (char *)&sin.sin_addr, hp->h_length);
    sin.sin_port = htons(SERVER_PORT);



    /* Open a TCP socket and assign handle new_s
       check for error; print message and exit if error
       if socket is successfully created, print message
       confirming the same.

       if socket is successfully created, connect to the
       remote host and print message if successful

       If connect fails, close the socket, print error message and exit.

       Function calls hints:
       socket(int socket_family, int socket_type, int protocol);

       int connect(int sockfd, const struct sockaddr *addr,
                   socklen_t addrlen)
     */
    // if (new_s = ... )


    /* 1. Implement the following protocol.
       2. Print all sent and received messages on screen.
       3. Your grades will be sent by the test server.
       4. You can assume that messages from server are
          at-most 256 bytes at a time

       Message exchange sequence to be implemented after open()
       ---------------------------------------------------------

       send "start"
       receive and print message from server
       send integer 4567890
       receive and print server message
       send "bye"
       receive and print message from server
       close the socket
    */


    while(1){} //remove it after completing the assignment
```

```
    return 0;
}
```