```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <stdio.h>
#include <sys/types.h>

// GROUP F
// RYAN ISENNOCK
// risenno@ostatemail.okstate.edu

int main(int argc, char *argv[])
{
   int sockfd;
   int newsockfd;
   int portNumber;
   int clientLength;
   int num;

   char name[256];
   char jobTitle[256];
   char status[256];

   struct sockaddr_in serverAddress, clientAddress;
   if(argc < 2)
   {
      exit(1);
   }

   sockfd = socket(AF_INET, SOCK_STREAM, 0);
```

```
if(sockfd < 0)
{
   return 1;
}

bzero((char *) &serverAddress, sizeof(serverAddress));
portNumber = atoi(argv[1]);

serverAddress.sin_family = AF_INET;
serverAddress.sin_addr.s_addr = INADDR_ANY;
serverAddress.sin_port = htons(portNumber);

if(bind(sockfd, (struct sockaddr *) &serverAddress, sizeof(serverAddress)) < 0)
{
   return 6;
}

//Only allowing up to two connections at a time
listen(sockfd, 2);

clientLength = sizeof(clientAddress);
newsockfd = accept(sockfd, (struct sockaddr *) &clientAddress, &clientLength);

if(newsockfd < 0)
{
   return 2;
}

int counter = 0;
```

```
//Receiving input from user up to a fixed amount of times. 100 in this case.
while(counter < 100){
   //Emptying variables every time
   bzero(name, 256);
   bzero(jobTitle, 256);
   bzero(status, 256);

   //Reading input
   num = read(newsockfd, name, 255);
   if(num < 0)
   {
      return 3;
   }

   //Reading input
   num = read(newsockfd, jobTitle, 255);
   if(num < 0)
   {
      return 4;
   }

   //Reading input
   num = read(newsockfd, status, 255);
   if(num < 0)
   {
      return 5;
   }
```

```
        //Calling assistant
        Assistant(name, jobTitle, status);
        counter++;
    }
    return 0;
}
```