Advanced Socket Programming

Write a program to send a message from server to client. Use TCP protocol.

```c
// tcps.c

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
void main() {
    struct sockaddr_in server, client;
    char buffer[100];
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
    client.sin_family = AF_INET;
    client.sin_addr.s_addr = INADDR_ANY;
    client.sin_port = 8080;
    bind(sockfd, (struct sockaddr*)&server, sizeof(server));
    listen(sockfd, 5);
    socklen_t len = sizeof(client);
    int clientfd = accept(sockfd, (struct sockaddr*)&client, &len);
    printf("Enter data to be sent: ");
    fgets(buffer, 100, stdin);
    send(clientfd, buffer, 100, 0); // server sends the message
    close(clientfd);
}
```

```c
// tcpc.c

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
void main() {
    struct sockaddr_in server;
    char buffer[100];
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
```

```
    connect(sockfd, (struct sockaddr*)&server, sizeof(server));
    recv(sockfd, buffer, 100, 0); // client receives the message
    printf("Message received is: %s", buffer);
    close(sockfd);
}
```

Write a socket program to send 5 integers to the server. Let the server compute the sum and send the result back to the client. Use TCP protocol.

```
// tcpc.c

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

void main() {
    struct sockaddr_in server, client;
    int i, arr[5], sum;
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
    client.sin_family = AF_INET;
    client.sin_addr.s_addr = INADDR_ANY;
    client.sin_port = 8080;
    bind(sockfd, (struct sockaddr*)&server, sizeof(server));
    listen(sockfd, 5);
    socklen_t len = sizeof(client);
    int clientfd = accept(sockfd, (struct sockaddr*)&client, &len);

    /*
    Important - here unlike char arrays, size of arr[5]
    is not 5. This is because each integer inside the
    array uses 4 bytes of memory, not 1 byte each. So use
    sizeof(arr) to get the total size. See below
    */

    recv(clientfd, arr, sizeof(arr), 0); // careful
    for(i=0; i<5; i++) {
        sum = sum + arr[i];
    }

    /*
```

```
    Important - arrays can be passed directly while
    sending or receiving. However for integers use &
    See below
    */

    send(clientfd, &sum, sizeof(sum), 0); // careful
    close(clientfd);
}

// tcpc.c

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

void main() {
    struct sockaddr_in server;
    int i, arr[5], sum;
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
    connect(sockfd, (struct sockaddr*)&server, sizeof(server));
    printf("Enter the numbers: ");
    for(i=0; i<5; i++) {
        scanf("%d", &arr[i]);
    }
    send(sockfd, arr, sizeof(arr), 0);  // careful
    recv(sockfd, &sum, sizeof(sum), 0); // careful
    printf("The sum is %d", sum);
    close(sockfd);
}
```

OUTPUT

```
sanbabyfrancis@DESKTOP-F4PQ7B4:/lenovo/programs/network$ gcc tcps.c -o serv
tcps.c: In function ‘main’:
tcps.c:24:5: warning: implicit declaration of function ‘close’; did you mea
   24 |     close(clientfd);
      |     ^~~~~
      |     pclose
sanbabyfrancis@DESKTOP-F4PQ7B4:/lenovo/programs/network$ ./server

sanbabyfrancis@DESKTOP-F4PQ7B4:/lenovo/programs/network$ gcc tcpc.c -o clie
tcpc.c: In function ‘main’:
```

```
tcpc.c:20:5: warning: implicit declaration of function ‘close’; did you mea
   20 |     close(sockfd);
      |     ^~~~~
      |     pclose
sanbabyfrancis@DESKTOP-F4PQ7B4:/lenovo/programs/network$ ./client
Enter the numbers: 1
2
3
4
5
The sum is 15
```

Write a socket program to send a message to server in uppercase. Let the server send the message back to client in lowercase. Use TCP protocol.

```c
// tcps.c

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include <ctype.h>

void main() {
    struct sockaddr_in server, client;
    int i;
    char buffer[100], lowermsg[100];
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
    client.sin_family = AF_INET;
    client.sin_addr.s_addr = INADDR_ANY;
    client.sin_port = 8080;
    bind(sockfd, (struct sockaddr*)&server, sizeof(server));
    listen(sockfd, 5);
    socklen_t len = sizeof(client);
    int clientfd = accept(sockfd, (struct sockaddr*)&client, &len);
    recv(clientfd, buffer, 100, 0);
    for(i=0; i<strlen(buffer); i++) {
        lowermsg[i] = tolower(buffer[i]);
    }
    send(clientfd, lowermsg, 100, 0);
```

```
    close(clientfd);
}

// tcpc.c

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

void main() {
    struct sockaddr_in server;
    char buffer[100], lowermsg[100];
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = INADDR_ANY;
    server.sin_port = 8080;
    connect(sockfd, (struct sockaddr*)&server, sizeof(server));
    printf("Enter message in uppercase to be sent: ");
    fgets(buffer, 100, stdin);
    send(sockfd, buffer, 100, 0);
    recv(sockfd, lowermsg, 100, 0);
    printf("The message received in lowercase is: %s", lowermsg);
    close(sockfd);
}
```

OUTPUT

```
sanbabyfrancis@DESKTOP-F4PQ7B4:/lenovo/programs/network$ gcc tcps.c -o serv
tcps.c: In function ‘main’:
tcps.c:27:5: warning: implicit declaration of function ‘close’; did you mea
   27 |     close(clientfd);
      |     ^~~~~
      |     pclose
sanbabyfrancis@DESKTOP-F4PQ7B4:/lenovo/programs/network$ ./server

sanbabyfrancis@DESKTOP-F4PQ7B4:/lenovo/programs/network$ gcc tcpc.c -o clie
tcpc.c: In function ‘main’:
tcpc.c:19:5: warning: implicit declaration of function ‘close’; did you mea
   19 |     close(sockfd);
      |     ^~~~~
      |     pclose
sanbabyfrancis@DESKTOP-F4PQ7B4:/lenovo/programs/network$ ./client
Enter message in uppercase to be sent: HELLO WORLD
The message received in lowercase is: hello world
```