

REPORT



과 목 :	네트워크프로그래밍
제출일자 :	2021. 10. 04
담당교수 :	황성호
학 과 :	컴퓨터공학과
학 번 :	201720970
이 름 :	권대한

1-1. TCP Server 고정길이 소스코드

```
#define _WINSOCK_DEPRECATED_NO_WARNINGS

#pragma comment(lib, "ws2_32")

#include <stdio.h>
#include <stdlib.h>
#include <WinSock2.h>

#define SERVERPORT 9000
#define BUFSIZE 50

// 소켓 함수 오류 출력 후 종료
void err_quit(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(
        FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
    MessageBox(NULL, (LPCTSTR)lpMsgBuf, msg, MB_ICONERROR);
    LocalFree(lpMsgBuf);
    exit(1);
}

// 소켓 함수 오류 출력
void err_display(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
    printf("[%s] %s", msg, (char*)lpMsgBuf);
    LocalFree(lpMsgBuf);
}
```

// 사용자 정의 데이터 수신 함수

```
int recvn(SOCKET s, char* buf, int len, int flags)
```

```
{
    int received;
    char* ptr = buf;
    int left = len;

    while (left > 0)
    {
        received = recv(s, ptr, left, flags);
        if (received == SOCKET_ERROR)
        {
            return SOCKET_ERROR;
        }
        else if (received == 0)
        {
            break;
        }
        left -= received;
        ptr += received;
    }

    return (len - left);
}
```

```
int main(int argc, char* argv[])
```

```
{
    int retval;

    //WinSock init
    WSADATA wsa;
    if (WSAStartup(MAKEWORD(2, 2), &wsa) != 0)
    {
        return 1;
    }

    //Socket()
    SOCKET listen_sock = socket(AF_INET, SOCK_STREAM, 0);
    if (listen_sock == INVALID_SOCKET)
    {

```

```

        err_quit("socket()");
    }

    //Bind()
    SOCKADDR_IN serveraddr;
    ZeroMemory(&serveraddr, sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = htonl(INADDR_ANY);
    serveraddr.sin_port = htons(SERVERPORT);
    retval = bind(listen_sock, (SOCKADDR*)&serveraddr, sizeof(serveraddr));
    if (retval == SOCKET_ERROR)
    {
        err_quit("bind()");
    }

    //Listen()
    retval = listen(listen_sock, SOMAXCONN);
    if (retval == SOCKET_ERROR)
    {
        err_quit("listen()");
    }

    // 데이터 통신에 사용할 변수
    SOCKET client_sock;
    SOCKADDR_IN clientaddr;
    int addrlen;
    char buf[BUFSIZE + 1];

    while (1)
    {
        //accept()
        addrlen = sizeof(clientaddr);
        client_sock = accept(listen_sock, (SOCKADDR*)&clientaddr, &addrlen);
        if (client_sock == INVALID_SOCKET)
        {
            err_display("accept()");
            break;
        }

        // 접속한 클라이언트 정보 출력
        printf("\n [TCP 서버] 클라이언트 접속: IP 주소=%s, 포트 번호=%d\n",

```

```
inet_ntoa(clientaddr.sin_addr), ntohs(clientaddr.sin_port));
```

```
//클라이언트와 데이터 통신
```

```
while (1)
```

```
{
```

```
    //데이터 받기
```

```
    retval = recvn(client_sock, buf, BUFSIZE, 0);
```

```
    if (retval == SOCKET_ERROR)
```

```
    {
```

```
        err_display("recv()");
```

```
        break;
```

```
    }
```

```
    else if (retval == 0)
```

```
    {
```

```
        break;
```

```
    }
```

```
    //받은 데이터 출력
```

```
    buf[retval] = '\0';
```

```
    printf("[TCP/%s:%d] %s\n", inet_ntoa(clientaddr.sin_addr),
```

```
    ntohs(clientaddr.sin_port), buf);
```

```
}
```

```
//closesocket()
```

```
closesocket(client_sock);
```

```
printf("[TCP 서버] 클라이언트 종료: IP 주소=%s, 포트 번호=%d\n",
```

```
inet_ntoa(clientaddr.sin_addr), ntohs(clientaddr.sin_port));
```

```
}
```

```
//closesocket()
```

```
closesocket(listen_sock);
```

```
//원속 종료
```

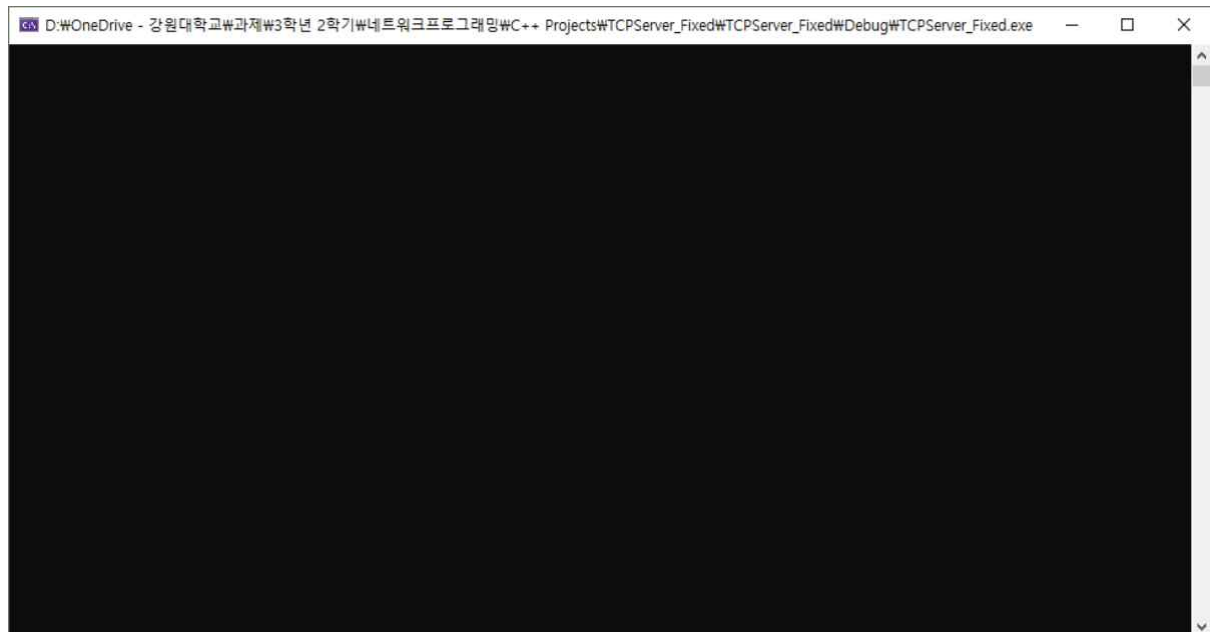
```
WSACleanup();
```

```
return 0;
```

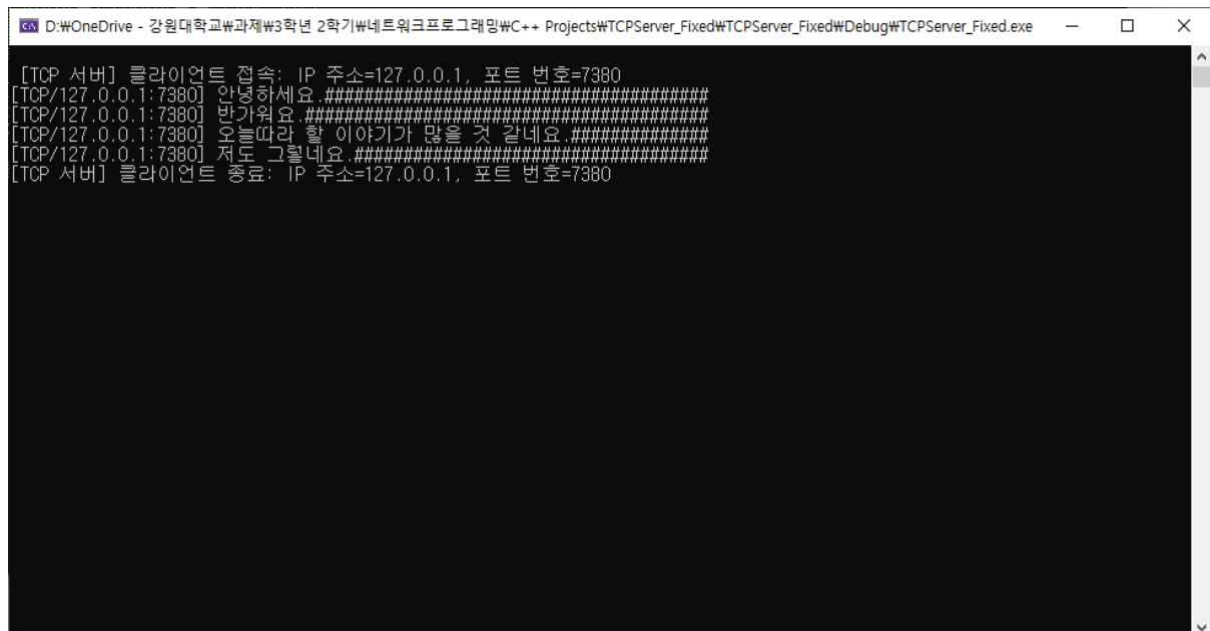
```
}
```

TCP Server 고정길이 출력 결과

Case 1. TCPServer 단독 실행 시



Case 2. TCPClient_Fixed 동시 실행 시



1-2. TCP Server 가변길이 소스코드

```
#define _WINSOCK_DEPRECATED_NO_WARNINGS

#pragma comment(lib, "ws2_32")

#include <stdio.h>
#include <stdlib.h>
#include <WinSock2.h>

#define SERVERPORT 9000
#define BUFSIZE 512

// 소켓 함수 오류 출력 후 종료
void err_quit(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(
        FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
    MessageBox(NULL, (LPCTSTR)lpMsgBuf, msg, MB_ICONERROR);
    LocalFree(lpMsgBuf);
    exit(1);
}

// 소켓 함수 오류 출력
void err_display(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
    printf("[%s] %s", msg, (char*)lpMsgBuf);
    LocalFree(lpMsgBuf);
}
```

```

int _recv_ahead(SOCKET s, char* p)
{
    __declspec(thread) static int nbytes = 0;
    __declspec(thread) static char buf[1024];
    __declspec(thread) static char* ptr;

    if (nbytes == 0 || nbytes == SOCKET_ERROR)
    {
        nbytes = recv(s, buf, sizeof(buf), 0);
        if (nbytes == SOCKET_ERROR)
        {
            return SOCKET_ERROR;
        }
        else if (nbytes == 0)
        {
            return 0;
        }
        ptr = buf;
    }

    --nbytes;
    *p = *ptr++;
    return 1;
}

```

```

int recvline(SOCKET s, char* buf, int maxlen)
{
    int n, nbytes;
    char c, *ptr = buf;

    for (n = 1; n < maxlen; n++)
    {
        nbytes = _recv_ahead(s, &c);
        if (nbytes == 1)
        {
            *ptr++ = c;
            if (c == '\n')
            {
                break;
            }
        }
    }
}

```



```

    }
    else if (nbytes == 0)
    {
        *ptr = 0;
        return n - 1;
    }
    else
    {
        return SOCKET_ERROR;
    }
}

*ptr = 0;
return n;
}

int main(int argc, char* argv[])
{

    int retval;

    //WinSock init
    WSADATA wsa;
    if (WSAStartup(MAKEWORD(2, 2), &wsa) != 0)
    {
        return 1;
    }

    //Socket()
    SOCKET listen_sock = socket(AF_INET, SOCK_STREAM, 0);
    if (listen_sock == INVALID_SOCKET)
    {
        err_quit("socket()");
    }

    //Bind()
    SOCKADDR_IN serveraddr;
    ZeroMemory(&serveraddr, sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = htonl(INADDR_ANY);
    serveraddr.sin_port = htons(SERVERPORT);

```

```

retval = bind(listen_sock, (SOCKADDR*)&serveraddr, sizeof(serveraddr));
if (retval == SOCKET_ERROR)
{
    err_quit("bind()");
}

//Listen()
retval = listen(listen_sock, SOMAXCONN);
if (retval == SOCKET_ERROR)
{
    err_quit("listen()");
}

// 데이터 통신에 사용할 변수
SOCKET client_sock;
SOCKADDR_IN clientaddr;
int addrlen;
char buf[BUFSIZE + 1];

while (1)
{
    //accept()
    addrlen = sizeof(clientaddr);
    client_sock = accept(listen_sock, (SOCKADDR*)&clientaddr, &addrlen);
    if (client_sock == INVALID_SOCKET)
    {
        err_display("accept()");
        break;
    }

    // 접속한 클라이언트 정보 출력
    printf("\n [TCP 서버] 클라이언트 접속: IP 주소=%s, 포트 번호=%d\n",
inet_ntoa(clientaddr.sin_addr), ntohs(clientaddr.sin_port));

    //클라이언트와 데이터 통신
    while (1)
    {
        //데이터 받기
        retval = recvline(client_sock, buf, BUFSIZE+1);
        if (retval == SOCKET_ERROR)
        {

```

```

        err_display("recv()");
        break;
    }
    else if (retval == 0)
    {
        break;
    }

    //받은 데이터 출력
    // 이미 사용자 지정함수에서 변환하였음...!
    //buf[retval] = '\0';
    printf("[TCP/%s:%d] %s", inet_ntoa(clientaddr.sin_addr),
    ntohs(clientaddr.sin_port), buf);
    }

    //closesocket()
    closesocket(client_sock);
    printf("[TCP 서버] 클라이언트 종료: IP 주소=%s, 포트 번호=%d\n",
    inet_ntoa(clientaddr.sin_addr), ntohs(clientaddr.sin_port));

    }

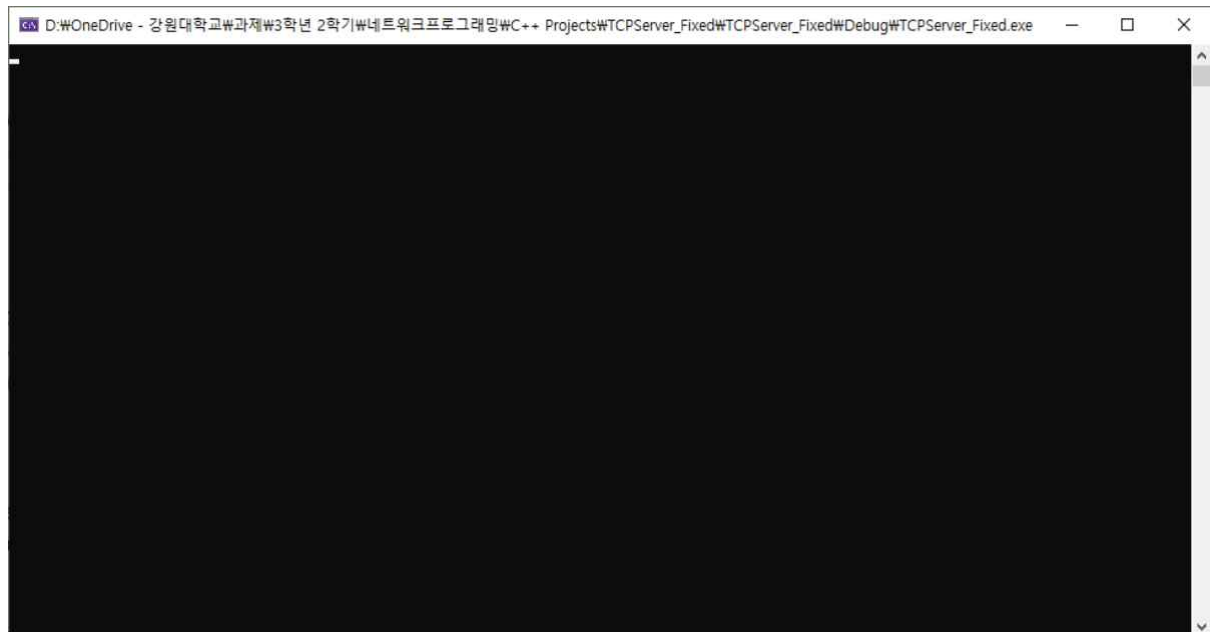
    //closesocket()
    closesocket(listen_sock);

    //원속 종료
    WSACleanup();
    return 0;
}

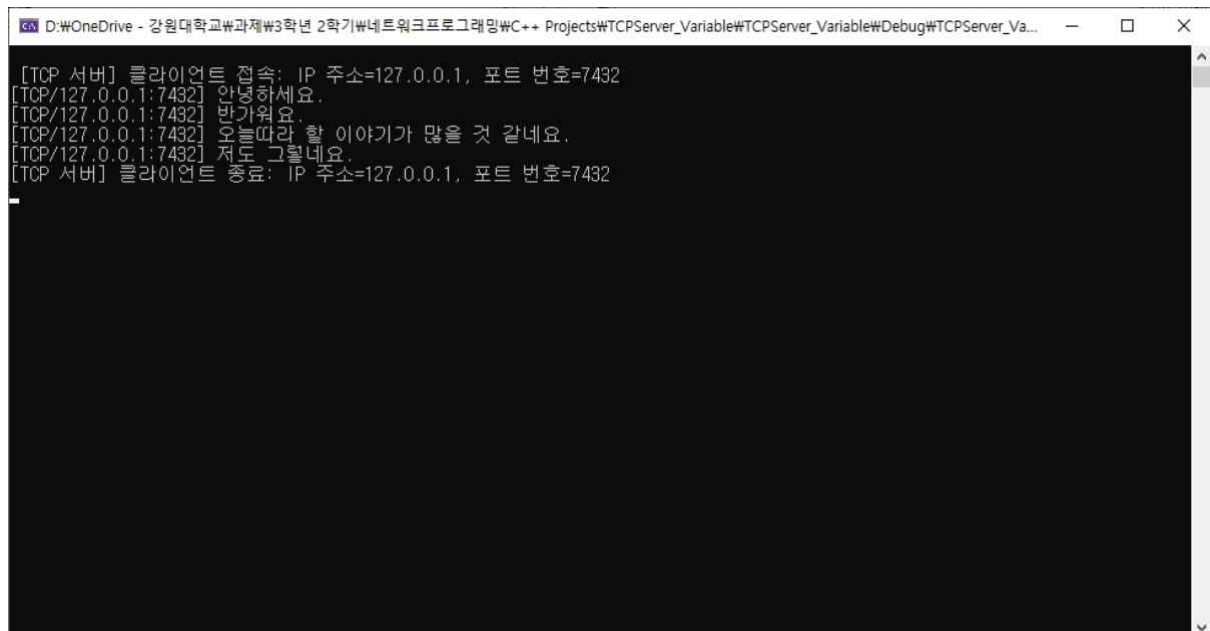
```

TCP Server 가변길이 출력 결과

Case 1. TCPServer 단독 실행 시



Case 2. TCPClient_Variable 동시 실행 시



2-1. TCP Client 고정 길이 소스코드

```
#pragma warning(disable : 4996)

#define _WINSOCK_DEPRECATED_NO_WARNINGS

#pragma comment(lib, "ws2_32")

#include <stdio.h>
#include <stdlib.h>
#include <WinSock2.h>

#define SERVERIP "127.0.0.1"
#define SERVERPORT 9000
#define BUFSIZE 50

// 소켓 함수 오류 출력 후 종료
void err_quit(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(
        FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
    MessageBox(NULL, (LPCTSTR)lpMsgBuf, msg, MB_ICONERROR);
    LocalFree(lpMsgBuf);
    exit(1);
}

// 소켓 함수 오류 출력
void err_display(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
    printf("[%s] %s", msg, (char*)lpMsgBuf);
}
```

```

        LocalFree(lpMsgBuf);
    }

int main(int argc, char* argv[])
{
    int retval;

    //원속 초기화
    WSADATA wsa;
    if (WSAStartup(MAKEWORD(2, 2), &wsa) != 0)
    {
        return 1;
    }

    //Socket()
    SOCKET sock = socket(AF_INET, SOCK_STREAM, 0);
    if (sock == INVALID_SOCKET)
    {
        err_quit("socket()");
    }

    //Connect()
    SOCKADDR_IN serveraddr;
    ZeroMemory(&serveraddr, sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = inet_addr(SERVERIP);
    serveraddr.sin_port = htons(SERVERPORT);
    retval = connect(sock, (SOCKADDR*)&serveraddr, sizeof(serveraddr));
    if (retval == SOCKET_ERROR)
    {
        err_quit("connect()");
    }

    // 데이터 통신에 사용할 변수
    char buf[BUFSIZE];
    char* testdata[] = {
        "안녕하세요.",
        "반가워요.",
        "오늘따라 할 이야기가 많을 것 같네요.",
        "저도 그렇네요."
    };
};

```

```

// 서버와 데이터 통신
for (int i = 0; i < 4; i++)
{
    //데이터 입력(시뮬레이션)
    memset(buf, '#', sizeof(buf));
    strncpy(buf, testdata[i], strlen(testdata[i]));

    //데이터 보내기
    retval = send(sock, buf, BUFSIZE, 0);
    if (retval == SOCKET_ERROR)
    {
        err_display("send()");
        break;
    }
    printf("[TCP 클라이언트] %d바이트를 보냈습니다.\n", retval);
}

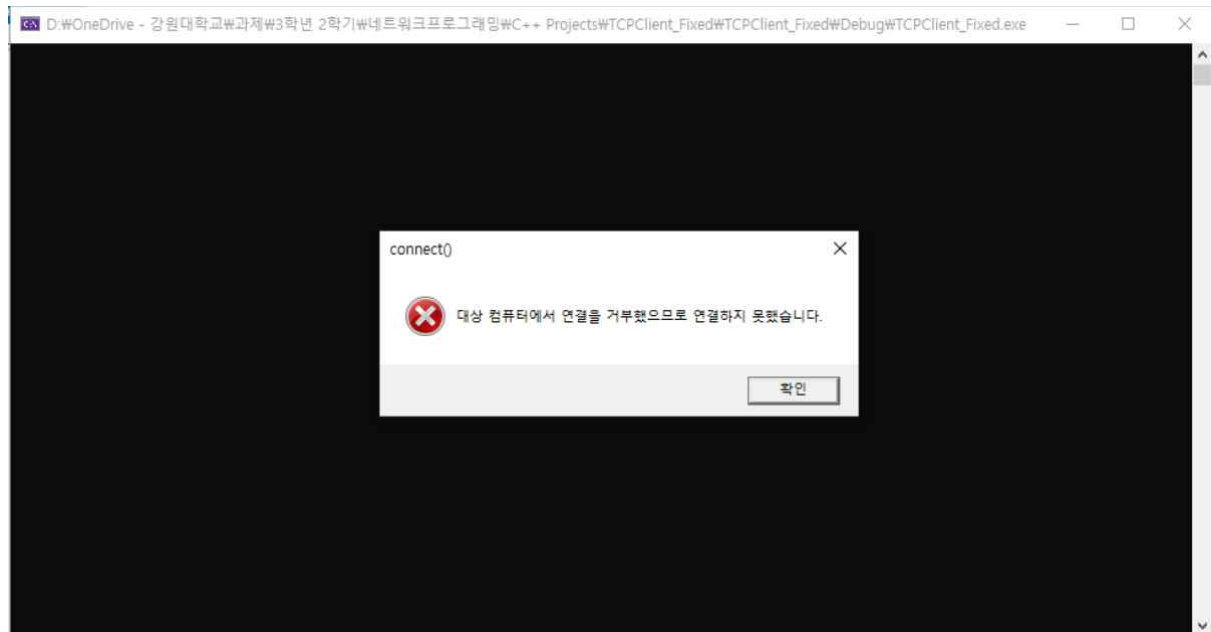
//Closesocket()
closesocket(sock);

//원속 종료
WSACleanup();
return 0;
}

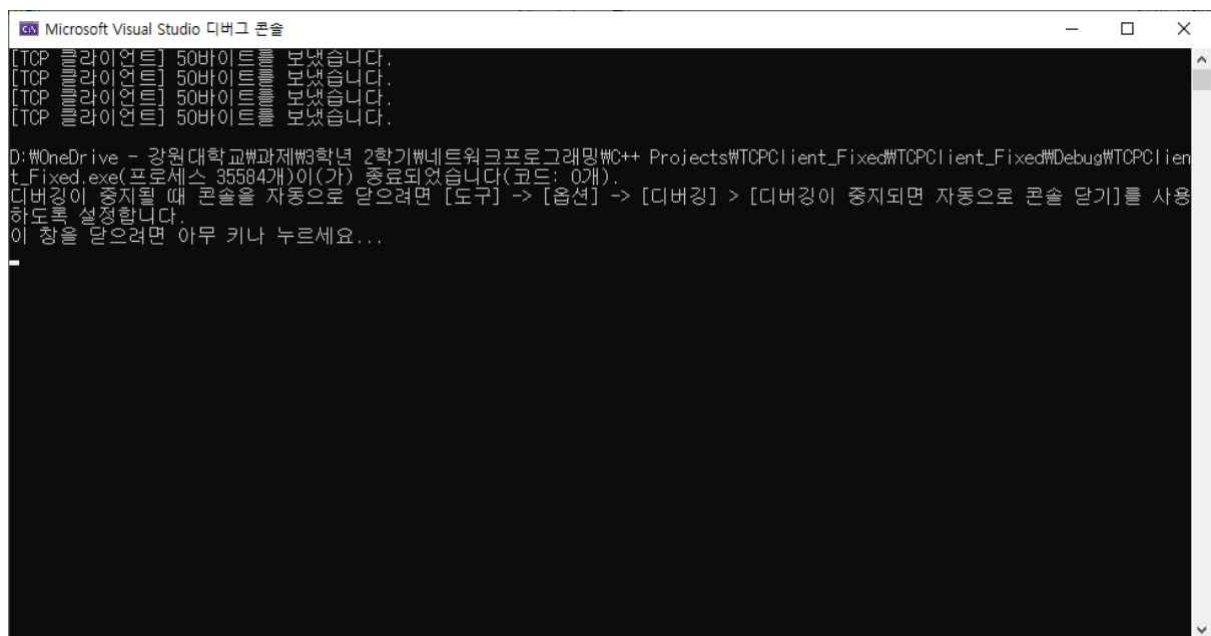
```

TCP Client 고정길이 출력 결과

Case 1. TCPClient_Fixed 단독 실행 시



Case 2. TCPServer_Fixed 동시 실행 시



2-2. TCP Client 가변길이 소스코드

```
#pragma warning(disable : 4996)

#define _WINSOCK_DEPRECATED_NO_WARNINGS

#pragma comment(lib, "ws2_32")

#include <stdio.h>
#include <stdlib.h>
#include <WinSock2.h>

#define SERVERIP "127.0.0.1"
#define SERVERPORT 9000
#define BUFSIZE 50

// 소켓 함수 오류 출력 후 종료
void err_quit(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(
        FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
    MessageBox(NULL, (LPCTSTR)lpMsgBuf, msg, MB_ICONERROR);
    LocalFree(lpMsgBuf);
    exit(1);
}

// 소켓 함수 오류 출력
void err_display(char* msg)
{
    LPVOID lpMsgBuf;
    FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_SYSTEM,
        NULL,
        WSAGetLastError(),
        MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
        (LPTSTR)&lpMsgBuf, 0, NULL);
    printf("[%s] %s", msg, (char*)lpMsgBuf);
}
```

```

        LocalFree(lpMsgBuf);
    }

int main(int argc, char* argv[])
{
    int retval;

    //원속 초기화
    WSADATA wsa;
    if (WSAStartup(MAKEWORD(2, 2), &wsa) != 0)
    {
        return 1;
    }

    //Socket()
    SOCKET sock = socket(AF_INET, SOCK_STREAM, 0);
    if (sock == INVALID_SOCKET)
    {
        err_quit("socket()");
    }

    //Connect()
    SOCKADDR_IN serveraddr;
    ZeroMemory(&serveraddr, sizeof(serveraddr));
    serveraddr.sin_family = AF_INET;
    serveraddr.sin_addr.s_addr = inet_addr(SERVERIP);
    serveraddr.sin_port = htons(SERVERPORT);
    retval = connect(sock, (SOCKADDR*)&serveraddr, sizeof(serveraddr));
    if (retval == SOCKET_ERROR)
    {
        err_quit("connect()");
    }

    // 데이터 통신에 사용할 변수
    char buf[BUFSIZE];
    char* testdata[] = {
        "안녕하세요.",
        "반가워요.",
        "오늘따라 할 이야기가 많을 것 같네요.",
        "저도 그렇네요."
    };
};

```

```

int len;

// 서버와 데이터 통신
for (int i = 0; i < 4; i++)
{
    //데이터 입력(시뮬레이션)
    len = strlen(testdata[i]);
    strncpy(buf, testdata[i], len);
    buf[len++] = '\n';

    //데이터 보내기
    retval = send(sock, buf, len, 0);
    if (retval == SOCKET_ERROR)
    {
        err_display("send()");
        break;
    }
    printf("[TCP 클라이언트] %d바이트를 보냈습니다.\n", retval);
}

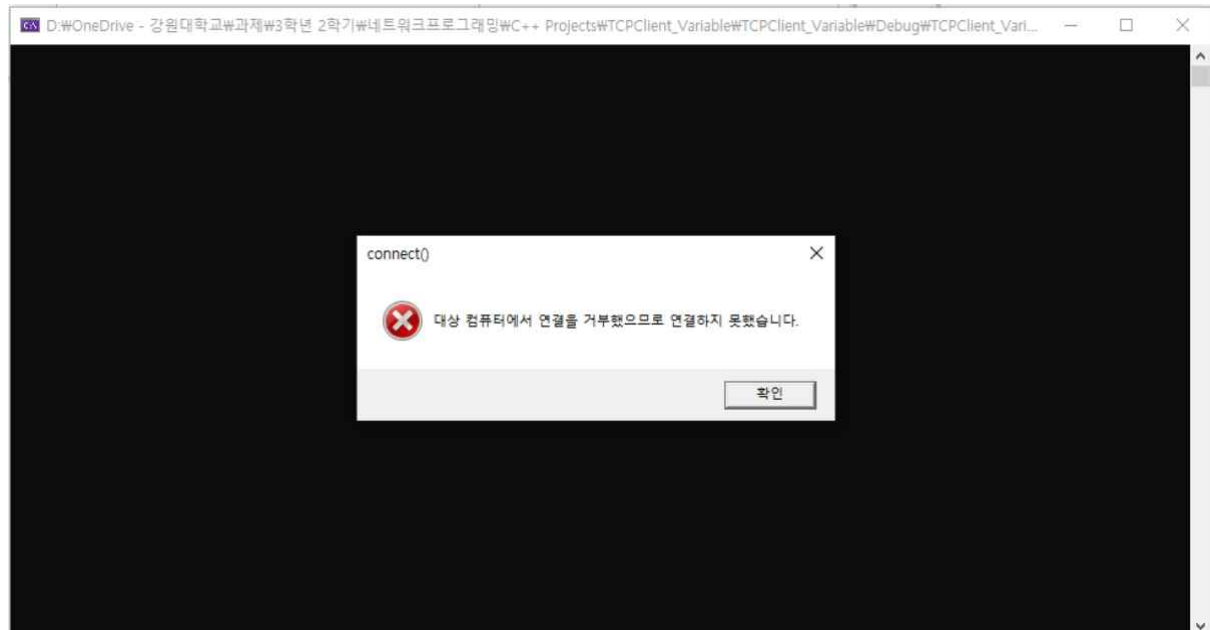
//Closesocket()
closesocket(sock);

//원속 종료
WSACleanup();
return 0;
}

```

TCP Client 가변길이 출력 결과

Case 1. TCPClient_Variable 단독 실행 시



Case 2. TCPServer_Variable 동시 실행 시

