```c
// This is the echo SERVER server.c
// cc -m32 server.c -o server
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <sys/socket.h>
#include <netdb.h>

#define  MAX 256

// Define variables:
struct sockaddr_in  server_addr, client_addr, name_addr;
struct hostent *hp;

int  sock, newsock;                  // socket descriptors
int  serverPort;                     // server port number
int  r, length, n;                   // help variables

// Server initialization code:

int server_init(char *name)
{
        printf("==================== server init ======================\n");
        // get DOT name and IP address of this host

        printf("1 : get and show server host info\n");
        hp = gethostbyname(name);
        if (hp == 0){
                printf("unknown host\n");
                exit(1);
        }
        printf("    hostname=%s  IP=%s\n",
               hp->h_name,  inet_ntoa(*(long *)hp->h_addr));

        //  create a TCP socket by socket() syscall
        printf("2 : create a socket\n");
        sock = socket(AF_INET, SOCK_STREAM, 0);
        if (sock < 0){
                printf("socket call failed\n");
                exit(2);
        }

        printf("3 : fill server_addr with host IP and PORT# info\n");
        // initialize the server_addr structure
        server_addr.sin_family = AF_INET;                   // for TCP/IP
        server_addr.sin_addr.s_addr = htonl(INADDR_ANY);   // THIS HOST IP address
        server_addr.sin_port = 0;   // let kernel assign port

        printf("4 : bind socket to host info\n");
        // bind syscall: bind the socket to server_addr info
        r = bind(sock,(struct sockaddr *)&server_addr, sizeof(server_addr));
        if (r < 0){
                printf("bind failed\n");
                exit(3);
        }

        printf("5 : find out Kernel assigned PORT# and show it\n");
        // find out socket port number (assigned by kernel)
        length = sizeof(name_addr);
        r = getsockname(sock, (struct sockaddr *)&name_addr, &length);
        if (r < 0){
                printf("get socketname error\n");
                exit(4);
        }

        // show port number
```

```
        serverPort = ntohs(name_addr.sin_port);   // convert to host ushort
        printf("    Port=%d\n", serverPort);

        // listen at port with a max. queue of 5 (waiting clients)
        printf("5 : server is listening ....\n");
        listen(sock, 5);
        printf("===================== init done ======================\n");
}


main(int argc, char *argv[])
{
        char *hostname;
        char line[MAX];

        if (argc < 2)
                hostname = "localhost";
        else
                hostname = argv[1];

        server_init(hostname);

        // Try to accept a client request
        while(1){
                printf("server: accepting new connection ....\n");

                // Try to accept a client connection as descriptor newsock
                length = sizeof(client_addr);
                newsock = accept(sock, (struct sockaddr *)&client_addr, &length);
                if (newsock < 0){
                        printf("server: accept error\n");
                        exit(1);
                }
                printf("server: accepted a client connection from\n");
                printf("-----------------------------------------------\n");
                printf("        IP=%s  port=%d\n", inet_ntoa(client_addr.sin_addr.s_addr),
                       ntohs(client_addr.sin_port));
                printf("-----------------------------------------------\n");

                // Processing loop
                while(1){
                        n = read(newsock, line, MAX);
                        if (n==0){
                                printf("server: client died, server loops\n");
                                close(newsock);
                                break;
                        }

                        // show the line string
                        printf("server: read  n=%d bytes; line=[%s]\n", n, line);

                        // Now we add the 2 numbers
                        char *string = strtok(line, " \n");
                        int a,b;
                        if(string != NULL){
                                printf("Var A '%s'\n", string);
                                a = strtol(string, (char **)NULL, 10);
                                string = strtok(NULL, " \n");
                                if(string != NULL && a != 0){
                                        printf("String '%s'\n", string);
                                        b = strtol(string, (char **)NULL, 10);
                                        if(b != 0){
                                                int c = a + b;
                                                sprintf(line, "Sum %d + %d = %d", a, b, c);
                                        }
                                }
                        }
```

```
//                      strcat(line, " ECHO"); didn't want to add this anymore

                        // send the echo line to client
                        n = write(newsock, line, MAX);

                        printf("server: wrote n=%d bytes; ECHO=[%s]\n", n, line);
                        printf("server: ready for next request\n");
                }
        }
}
```