

```

//===== file=getaddr.c =====Pseudo Code=====
//= A program to get the IP address for a given host name
=
//=====
==
//= Notes:
=
//= 1) This program conditionally compiles for Winsock and BSD sockets.
=
//= Set the initial #define to WIN or BSD as appropriate.
=
//= 2) This program assumes command line entry of the host name.
=
//-----
--
//= Example execution:
=
//=
=
//= getaddr computerassignmenthelp
=
//-----
--
//= Build: bcc32 getaddr.c or cl getaddr.c wsock32.lib for Winsock
=
//= gcc getaddr.c -lsocket -lnsl for BSD
=
//-----
--
//= Execute: getaddr host_name
=
//-----
--
//=====
==
#define WIN // WIN for Winsock and BSD for BSD sockets

//----- Include files -----
--
#include <stdio.h> // Needed for printf()
#include <stdlib.h> // Needed for exit()
#include <string.h> // Needed for memcpy() and strcpy()
#ifdef WIN
#include <windows.h> // Needed for all Winsock stuff
#endif
#ifdef BSD
#include <sys/types.h> // Needed for system defined identifiers.
#include <netinet/in.h> // Needed for internet address structure.
#include <arpa/inet.h> // Needed for "inet_ntoa".
#include <sys/socket.h> // Needed for socket(), bind(), etc...
#include <fcntl.h>
#include <netdb.h>
#endif

//===== Main program
=====
void main(int argc, char *argv[])

```

```

{
#ifdef WIN
    WORD wVersionRequested = MAKEWORD(1,1);          // Stuff for WSA functions
    WSADATA wsaData;                                // Stuff for WSA functions
#endif

    struct hostent *host;                            // Structure for gethostbyname()
    struct in_addr address;                          // Structure for Internet address
    char          host_name[256];                   // String for host name

    if (argc != 2)
    {
        printf("*** ERROR - incorrect number of command line arguments \n");
        printf("          usage is 'getaddr host_name' \n");
        exit(1);
    }

#ifdef WIN
    // This stuff initializes winsock
    WSASStartup(wVersionRequested, &wsaData);
#endif

    // Copy host name into host_name
    strcpy(host_name, argv[1]);

    // Do a gethostbyname()
    printf("Looking for IP address for '%s'... \n", host_name);
    host = gethostbyname(host_name);

    // Output address if host found
    if (host == NULL)
        printf("  IP address for '%s' could not be found \n", host_name);
    else
    {
        memcpy(&address, host->h_addr, 4);
        printf("  IP address for '%s' is %s \n", host_name, inet_ntoa(address));
    }

#ifdef WIN
    // This stuff cleans-up winsock
    WSACleanup();
#endif
}

```