```
struct addrinfo {
 int              ai_flags;
 int              ai_family;
 int              ai_socktype;
 int              ai_protocol;
 socklen_t        ai_addrlen;
 struct sockaddr *ai_addr;
 char            *ai_canon name;
 struct addrinfo *ai_next;
};
```

Client Review : What are the steps to setting up a client TCP socket?
1. 2. 3.

How many addrinfo structs does getaddrinfo return? Why?

How do I get a string error with getaddrinfo returns?

What is AF_INET6?

What is 0:0:0:0:0:0:0:1?

Using getaddrinfo wow would I request stream-based protocol for https?

```
int startserver() {
  struct addrinfo hints, *result;
  ?
  hints.ai_family  =
  hints.ai_socktype =

  int result = _________________(_________________,_____,_________,_________) ?

}
```

For each addrinfo what do you call next?

Can you bind() a client? Why would you want to?

TCP SERVER

What is a passive socket? How do you specify it?

Why would I create one?

If you don't *bind* what do you get?

What is htons? ntohs? Why/when do we need them?

```
struct sockaddr_in stSockAddr;
int SocketFD = socket(PF_INET, SOCK_STREAM, IPPROTO_TCP);

memset(&stSockAddr, 0, sizeof(stSockAddr));
stSockAddr.sin_family = AF_INET;
stSockAddr.sin_port = htons(1100);
stSockAddr.sin_addr.s_addr = htonl(INADDR_ANY);
```

What are the "four calls "? What is their order? And what is their purpose?

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <unistd.h>
#include <arpa/inet.h>

int main(int argc, char** argv)
{
        int s;
        int sock_fd = socket(AF_INET, SOCK_STREAM, 0);

        struct addrinfo hints, *result;
        memset(&hints, 0, sizeof(struct addrinfo));
        hints.ai_family = AF_INET;
        hints.ai_socktype = SOCK_STREAM;
        hints.ai_flags = AI_PASSIVE;

        s = getaddrinfo(NULL, "1234", &hints, &result);
        if (s != 0) {
             fprintf(stderr, "getaddrinfo: %s\n", gai_strerror(s));
        exit(1);
        }

        if ( bind(sock_fd, result->ai_addr, result->ai_addrlen) != 0 ) {
                perror("bind()"); exit(1);
        }
        if ( listen(sock_fd, 10) != 0 {
                perror("listen()"); exit(1);
        }

   struct sockaddr_in * result_addr = (struct sockaddr_in*) result->ai_addr;
   printf("Listening on file descriptor %d, port %d\n", sock_fd, ntohs(result_addr->sin_port));
     //inet_ntoa(result_addr->sin_addr),

        printf("Waiting for connection...\n");
        int client_fd = accept(sock_fd, NULL, NULL);
        printf("Connection made: client_fd=%d\n", client_fd);

        char buffer[1000];
        int len = read(client_fd, buffer, 999);
        buffer[len] = '\0';

        printf("Read %d chars\n", len);
        printf("===\n");
        printf("%s\n", buffer);

   return 0;
}
```

(What is a 'honey pot?)

What is epoll?