4. Write a server/client model socket program to exchange hello message between them.

Server Side program:

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <string.h>

#define PORT 8076

int main(){
    int server_fd, new_socket;
    ssize_t valread;
    struct sockaddr_in address;
    int opt = 1;
    socklen_t addrlen = sizeof(address);

    char buffer[1024]={ 0 };
    char* hello = "Hello from server";

    //creating sof=cket file descriptor
    if((server_fd = socket(AF_INET, SOCK_STREAM, 0 ))<0) {
        printf("Socket not connected");
        exit(1);
    }

    //optional
    if(setsockopt(server_fd, SOL_SOCKET,SO_REUSEADDR | SO_REUSEPORT, &opt,
sizeof(opt))){
        printf("server setscokopt not working");
        exit(1);
            }

    //connect socket to port and ip
    address.sin_family = AF_INET;
    address.sin_addr.s_addr = INADDR_ANY;
    address.sin_port = htons(PORT);

    if(bind(server_fd, (struct sockaddr*)&address, sizeof(address))<0){
        perror("bind failed");
        exit(1);
    }
```

```
    if(listen(server_fd,3)<0){
        perror("listen");
        exit(1);
    }

    if((new_socket = accept(server_fd, (struct sockaddr*)&address, &addrlen))<0){
        perror("accept");
        exit(1);
    }

    valread = read(new_socket, buffer,1023);
    printf("%s\n",buffer);

    send(new_socket,hello,strlen(hello),0);
    printf("hello msg sent from server...\n");

    close(new_socket);
    close(server_fd);
}
```

Client Side Program:

```
#include <arpa/inet.h>
#include <unistd.h>
#include <sys/socket.h>
#include <stdio.h>
#include <string.h>

#define PORT 8076


int main(int argc, char const* argv[]){
    char buffer[1024] = { 0 };
    struct sockaddr_in server_addr;
    int client_fd,val_read,status;
    char* msg = "hello from client";


    if((client_fd = socket(AF_INET,SOCK_STREAM,0)) < 0){
        printf("socket not created");
        return -1;
    }
```

```
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(PORT);

    if(inet_pton(AF_INET,"127.0.0.1",&server_addr.sin_addr)<=0){
        printf("Address is in incompatible type");
        return -1;
    }

    if((status = connect(client_fd,(struct
sockaddr*)&server_addr,sizeof(server_addr)))<0){
        printf("unable to connect!");
        return -1;
    }

    send(client_fd, msg,strlen(msg),0);
    printf("msg sent from client...\n");

   val_read = read(client_fd,buffer,1023);
   printf("%s\n",buffer);

   close(client_fd);
   return 0;

}
```

Output at Server side:

```
aswin@EURLTP-379:/mnt/c/Users/Aswin/Documents$ gcc hello_server.c -o hello_server
aswin@EURLTP-379:/mnt/c/Users/Aswin/Documents$ ./hello_server
hello from client
hello msg sent from server...
aswin@EURLTP-379:/mnt/c/Users/Aswin/Documents$
```

Output at Client side:

```
aswin@EURLTP-379:/mnt/c/Users/Aswin/Documents$ gcc hello_client.c -o hello_client
aswin@EURLTP-379:/mnt/c/Users/Aswin/Documents$ ./hello_client
msg sent from client...
Hello from server
aswin@EURLTP-379:/mnt/c/Users/Aswin/Documents$
```