UDP Client Socket Program:

```
#include <stdio.h>
#include <strings.h> //bzero()
#include <sys/types.h> //special data types used in socket programming
#include <arpa/inet.h> //to convert IPv4 address to dotted decimal format
#include <sys/socket.h> //to create socket and data transfer function
#include <netinet/in.h> // For sockaddr_in structure and htons()
#include <unistd.h>  //close() connection
#include <stdlib.h>

#define PORT 5000
#define MAXLINE 1000

int main()
{
    char buffer[100];
    char *message = "Hello Server";
    int n;
    struct sockaddr_in servaddr;  //Structure to hold the server's address
information

   // clear servaddr structure
    bzero(&servaddr, sizeof(servaddr));
   //Initialize sockaddr
    servaddr.sin_family = AF_INET;  //IPv4 Protocol used
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");  //local server IP
    servaddr.sin_port = htons(PORT);  //convert to network byte order

   // create datagram socket
    int sockfd = socket(AF_INET, SOCK_DGRAM, 0);

    // request to send datagram
    sendto(sockfd, message, MAXLINE, 0, (struct sockaddr*)&servaddr,
sizeof(servaddr));
    // waiting for response
    recvfrom(sockfd, buffer, sizeof(buffer), 0, NULL, NULL);
    printf("\nReceived from Server:%s",buffer);
    // close the file descriptor
    close(sockfd);
}
```

UDP Server Socket Program:

```
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <stdlib.h>

#define PORT 5000
#define MAXLINE 1000

int main()
{
    char buffer[100];
    char *message = "Hello Client";
    socklen_t len;
    struct sockaddr_in servaddr, cliaddr;
    bzero(&servaddr, sizeof(servaddr));

   // Create a UDP Socket
    int serverfd = socket(AF_INET, SOCK_DGRAM, 0);
    //Initialize sockaddr
    servaddr.sin_family = AF_INET;   //IPv4 Protocol used
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY); //local IP address
    servaddr.sin_port = htons(PORT); //convert to network byte order

    // bind server address to socket descriptor
    bind(serverfd, (struct sockaddr*)&servaddr, sizeof(servaddr));
    printf("\nServer Running .........\n");

    //receive the datagram
    len = sizeof(cliaddr);
    int n = recvfrom(serverfd, buffer, sizeof(buffer),
            0, (struct sockaddr*)&cliaddr, &len);
    buffer[n] = '\0';
    printf("\nReceived from Client: %s\n",buffer);
    sendto(serverfd, message, MAXLINE, 0,
          (struct sockaddr*)&cliaddr, sizeof(cliaddr));
    printf("\nMessage sent to client");

    close(serverfd);
    return 0;
}
```