NAME

MOHAMMAD SHAAD

REGISTRATION NUMBER

21BCE1542

CLASS

COMPUTER NETWORKS

FACULTY

PUNITHA K MA'AM

LAB

EXERCISE 8

Q. Implement a client/server program using UDP for the following

- Client sends a message to server
- Server encodes the message by replacing a character by the charater next to it

    for example a to b, c to d

- Display the message and output on the client side

Write the code in c language for the server and client

Code

server.c

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

#define PORT 10000

int main() {
    int sockfd;
    struct sockaddr_in servaddr, cliaddr;
    socklen_t len = sizeof(cliaddr);
    char buffer[1024];

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

    memset(&servaddr, 0, sizeof(servaddr));
```

```c
    // fill server information
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = INADDR_ANY;
    servaddr.sin_port = htons(PORT);

    // bind the socket with the server address
    bind(sockfd, (const struct sockaddr *)&servaddr, sizeof(servaddr));

    while (1) {
        // receive data from a client
        int n = recvfrom(sockfd, (char *)buffer, 1024, MSG_WAITALL, (struct sockaddr *)&cliaddr, &len);
        buffer[n] = '\0';

        // encode the message from the client
        char encoded_message[1024];
        for(int i = 0; i < strlen(buffer); i++) {
            if(buffer[i] >= 'a' && buffer[i] <= 'z') {
                encoded_message[i] = buffer[i] == 'z' ? 'a' : buffer[i]+1;
            } else if (buffer[i] >= 'A' && buffer[i] <= 'Z') {
                encoded_message[i] = buffer[i] == 'Z' ? 'A' : buffer[i]+1;
            } else {
                encoded_message[i] = buffer[i];
            }
        }

        printf("Received message: %s\n", buffer);
        printf("Encoded message: %s\n", encoded_message);

        // send the encoded message back to the client
        sendto(sockfd, (const char *)encoded_message, strlen(encoded_message), MSG_CONFIRM, (const struct sockaddr *)&cliaddr, len);
    }
```

```
    return 0;
}
```

Client.c

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

#define PORT 10000

int main() {
    int sockfd;
    struct sockaddr_in servaddr;

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

    memset(&servaddr, 0, sizeof(servaddr));

    // fill server information
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(PORT);
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");

    // ask for user input
    char message[1024];
    printf("Enter a message: ");
    fgets(message, 1024, stdin);
```

```c
    message[strcspn(message, "\n")] = 0;


    // send the message to the server
    sendto(sockfd, (const char *)message, strlen(message), MSG_CONFIRM, (const struct sockaddr
*)&servaddr, sizeof(servaddr));
    printf("Sent message: %s\n", message);


    // receive the encoded message from the server
    char buffer[1024];
    socklen_t len = sizeof(servaddr);
    int n = recvfrom(sockfd, (char *)buffer, 1024, MSG_WAITALL, (struct sockaddr *)&servaddr,
&len);
    buffer[n] = '\0';
    printf("Received encoded message: %s\n", buffer);


    // close the socket
    close(sockfd);


    return 0;
}
```

Output