Exercise 5(UDP)

Aim:

To find whether the given number is odd or even using UDP protocol

C Program:

Server:

```
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<unistd.h>
#include<netdb.h>
#include<stdio.h>
#include<string.h>
#include<arpa/inet.h>
#define MAXLINE 1024
int main()
{
int sockfd;
int n,a,b;
socklen_t len;
char message[1024];
struct sockaddr_in servaddr,cliaddr;

//Socket creation
sockfd=socket(AF_INET,SOCK_DGRAM,0);
if(sockfd==-1)
{
        printf("Socket Creation Failed..\n");
}
else
{
        printf("Socket created Successfully..\n");
}
```

```
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=INADDR_ANY;
servaddr.sin_port=htons(5035);

//Binding process
b=bind(sockfd,(struct sockaddr*)&servaddr,sizeof(servaddr));
if(b==-1)
{
        printf("Binding Failed..\n");
}
else
{
        printf("Binded Successfully..\n");
}
len=sizeof(cliaddr);
while(1)
{
      n=recvfrom(sockfd,message,MAXLINE,0,(struct sockaddr*)&cliaddr,&len);
printf("Client's Message : %s\n",message);
        if(strcmp(message,"bye")==0)
        {
          strcpy(message,"bye");
            printf("Server Message : %s\n",message);
            printf("Server exit..\n");
            sendto(sockfd,message,n,0,(struct sockaddr*)&cliaddr,len);
            break;
        }
        else
        {
                a=atoi(message);
                if(a%2==0)
                {
                        printf("Server Message:");
                        printf("\t Even Number\n");
                        strcpy(message,"Even Number");
                        sendto(sockfd,message,n,0,(struct sockaddr*)&cliaddr,len);
                }
                else
                {
```

```
                        printf("Server Message:");
                        printf("\t Odd Number\n");
                        strcpy(message,"Odd Number");
                        sendto(sockfd,message,n,0,(struct sockaddr*)&cliaddr,len);
                }

        }
}

return 0;
}
```

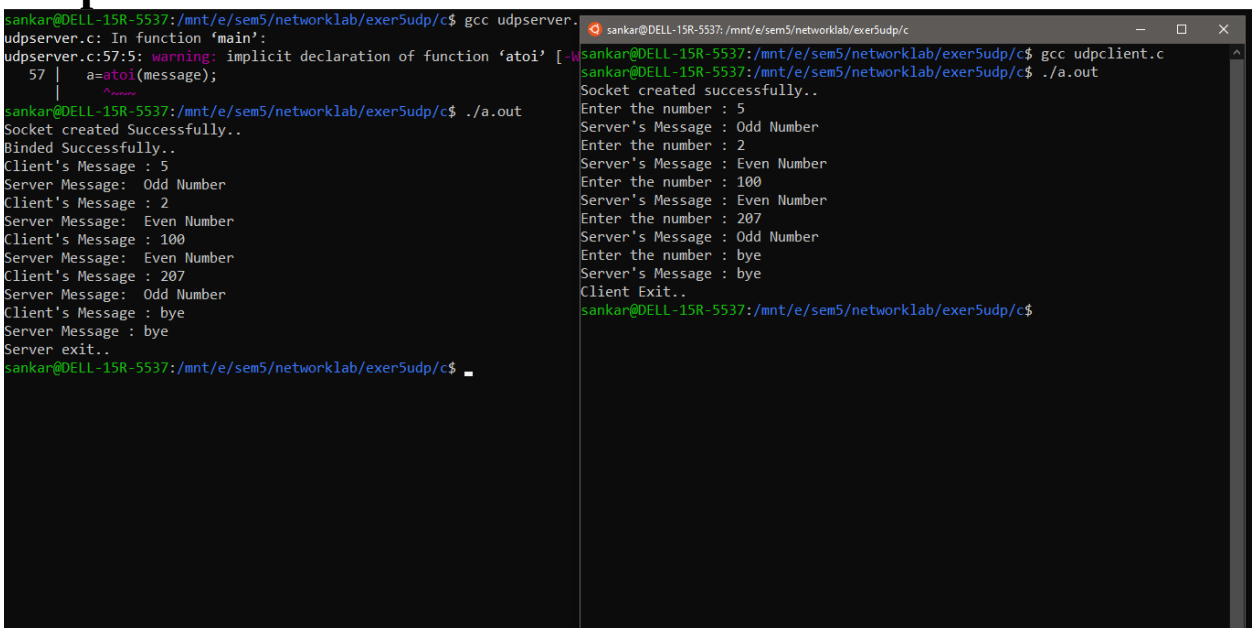
Client:

```
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include<arpa/inet.h>
#include<string.h>
#include<arpa/inet.h>
#include<stdio.h>
#define MAXLINE 1024
int main()
{
int sockfd;
int n;
socklen_t len;
char cliemsg[1024],servmsg[1024];
struct sockaddr_in servaddr;
sockfd=socket(AF_INET,SOCK_DGRAM,0);
if(sockfd==-1)
{
        printf("Socket creation failed..\n");
}
else
{
        printf("Socket created successfully..\n");
}
```

```
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=inet_addr("127.0.0.1");
servaddr.sin_port=htons(5035);
len=sizeof(servaddr);
while(1)
{
        strcpy(cliemsg,"");
        printf("Enter the number : ");
        scanf("%s",cliemsg);
        sendto(sockfd,cliemsg,MAXLINE,0,(struct sockaddr*)&servaddr,len);
        n=recvfrom(sockfd,servmsg,MAXLINE,0,NULL,NULL);
        servmsg[n]=0;
        printf("Server's Message : %s\n",servmsg);
        if(strcmp(servmsg,"bye")==0)
        {
                printf("Client Exit..\n");
                break;
        }

}
return 0;
}
```

Output:



Java Program:

Server:

```
import java.net.*;
import java.io.*;
import java.util.Scanner;
public class udpserver
{

    private int port;
        private Socket sd=null;
        private ServerSocket server= null;
        private DataInputStream in=null;

        public udpserver(int port)
        {
                try
                {
                  //creation of socket
                        server=new ServerSocket(port);
                        System.out.println("Server started");
                        System.out.println("Waiting for a client ...");
```

```
                    //accepting client request
                    sd = server.accept();
                System.out.println("Client accepted");


               // takes input from the client socket
            in = new DataInputStream(new
BufferedInputStream(sd.getInputStream()));
                     PrintWriter out=new PrintWriter(sd.getOutputStream(),true);
                String message = "";
               // reads message from client until "bye" is sent
               while (true)
                  {
                     try
                     {
                      message = in.readUTF();
                     System.out.println("Client Message : "+message);
                            if(message.equals("bye"))
                            {
                                  System.out.println("Server Message : "+message);
                                  out.println(message);
                                break;
                            }
                     else
```

```
                    {
                    int i = Integer.parseInt(message);
                     if(i%2==0)
                            {
                                    System.out.println("Server Message : Even
Number");
                                    message="Even Number";
                             //send result to client
                            out.println(message);
                            }
                            else
                            {
                            System.out.println("Server Message : Odd
Number");
message="Odd Number";
//send result to client
                            out.println(message);
                    }
                }
            }
        catch(Exception i){ System.out.println(i); }
      }
    System.out.println("Server exit..");
    sd.close();
```

```
        }
      catch(Exception i){ System.out.println(i); }
  }
   public static void main(String args[])
   {
      Scanner input=new Scanner(System.in);
      System.out.print("Enter port no : ");
      int portno=input.nextInt();
      udpserver udp=new udpserver(portno);
   }
}
```

Client:

```
import java.net.*;
import java.io.*;
import java.util.Scanner;

public class udpclient
{
        private static final String Server_IP="127.0.0.1";
        private int port;
        private Socket sd=null;
```

```
  private ServerSocket server=null;
  private DataInputStream input= null;
  private DataOutputStream out = null;
  udpclient(int port)
  {
          try
          {
                  sd=new Socket(Server_IP,port);
                  System.out.println("Connected");
                  // takes input from terminal
                  input = new DataInputStream(System.in);

                  // sends output to the socket
                  out = new DataOutputStream(sd.getOutputStream());
  }
  catch(Exception i){ System.out.println(i); }
     String climsg = "";
// keep reading until "bye" is input
  while (true)
          {
                  try
                  {
```

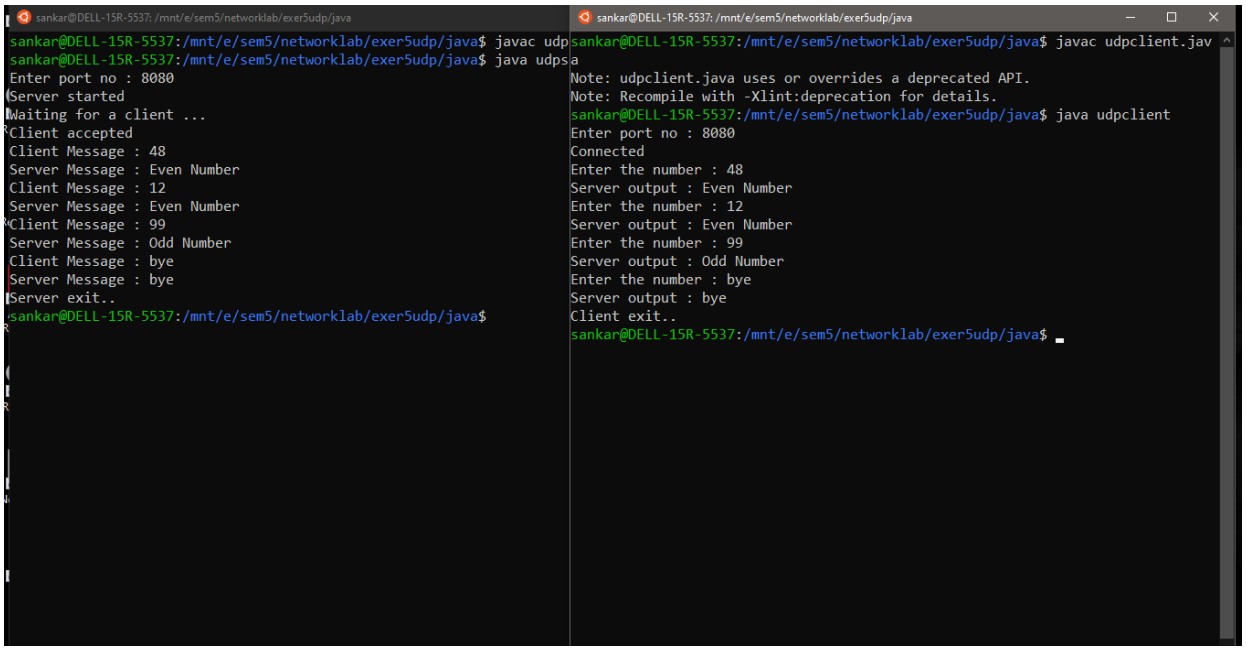
```
                    System.out.print("Enter the number : ");
climsg = input.readLine();
//send input data to server for processing
                    out.writeUTF(climsg);


                    //read input got from server
                    BufferedReader ip=new BufferedReader(new
InputStreamReader(sd.getInputStream()));
String servermessage=ip.readLine();
System.out.println("Server output : "+servermessage);
                    if(servermessage.equals("bye"))
{
                        System.out.println("Client exit..");
                        break;
        }
                }
              catch(IOException i){ System.out.println(i); }
          }
            try
            {
                input.close();
                out.close();
                sd.close();
           }
         catch(IOException i){ System.out.println(i); }
```

```
    }

    public static void main(String args[])
    {
        Scanner input=new Scanner(System.in);
        System.out.print("Enter port no : ");
        int portno=input.nextInt();
        udpclient udp=new udpclient(portno);
    }
}
```

Output:



Result:

Thus the above programs were executed successfully.