Name: Preyash Date: 10-02-2022

Registration Number: 20BPS1022

LAB-06

UDP Socket Programming

AIM: To compute the sum of the first 'n' even numbers, write a UDP socket application and run it. The server computes the sum using the 'n' value provided by the client, and the result should be shown on the client side.

ALGORITHM

Server-side Algorithm:

- ✓ Start
- ✓ Include the sys/socket.h> and arpa/inet.h header files.
- ✓ Using the socket() method, create a socket that returns a socket descriptor.
- ✓ Set the server address based on the port and IP.
- ✓ Bind the socket to the server address with bind().
- ✓ Using the recvfrom() method, get the value of "n" from the client. Then, using n*(n+1), calculate the value of the sum of "n" even integers.
- ✓ Send the result back to the client.
- ✓ End.

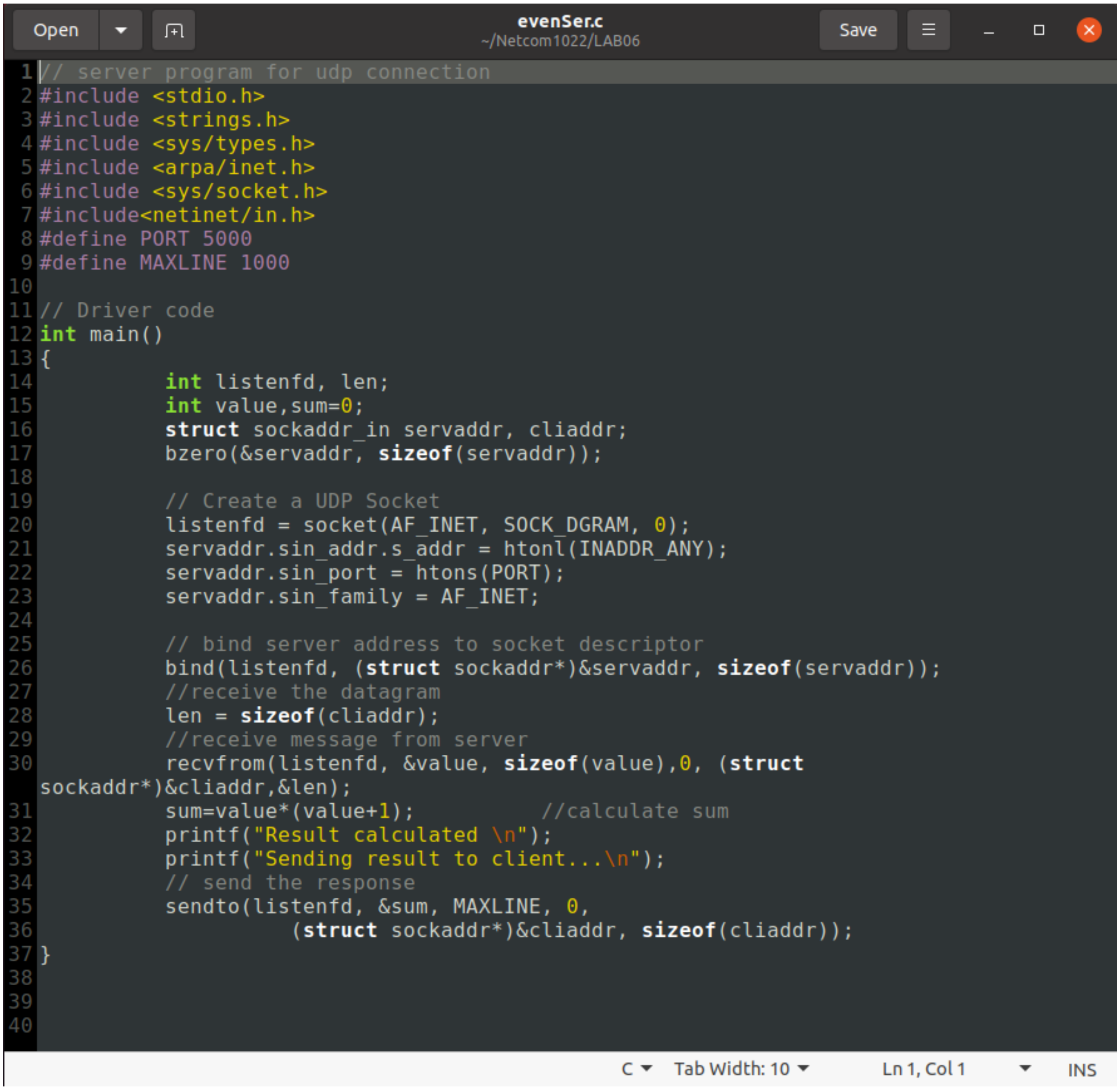
Client-Side Algorithm:

- ✓ Start
- ✓ Include the sys/socket.h> and arpa/inet.h header files.
- ✓ Create a datagram socket and use the connect() function to connect to the server.
- ✓ Take the user's value for "n."
- ✓ If you're connected, send the value to the server.

- ✓ Using the recvfrom() method, wait for a response from the server.
- ✓ Obtain the server's result and print it.
- ✓ To cease the conversation, close the socket.
- ✓

Server Program Source Code:

Code window:

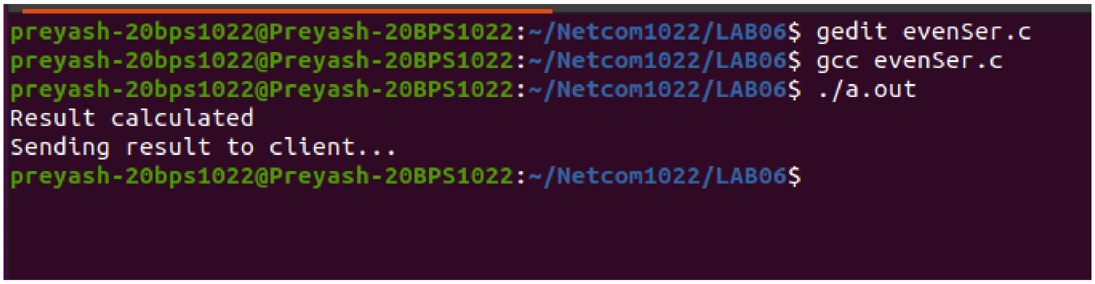


```
// server program for udp connection
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include<netinet/in.h>
#define PORT 5000
#define MAXLINE 1000

// Driver code
int main()
{
          int listenfd, len;
          int value,sum=0;
          struct sockaddr_in servaddr, cliaddr;
          bzero(&servaddr, sizeof(servaddr));

          // Create a UDP Socket
          listenfd = socket(AF_INET, SOCK_DGRAM, 0);
          servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
          servaddr.sin_port = htons(PORT);
          servaddr.sin_family = AF_INET;

          // bind server address to socket descriptor
          bind(listenfd, (struct sockaddr*)&servaddr, sizeof(servaddr));
          //receive the datagram
          len = sizeof(cliaddr);
          //receive message from server
          recvfrom(listenfd, &value, sizeof(value),0, (struct
sockaddr*)&cliaddr,&len);
          sum=value*(value+1);          //calculate sum
          printf("Result calculated \n");
          printf("Sending result to client...\n");
          // send the response
          sendto(listenfd, &sum, MAXLINE, 0,
                    (struct sockaddr*)&cliaddr, sizeof(cliaddr));
}
```

Code:

```
// server program for udp connection
#include <stdio.h>
#include <strings.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include<netinet/in.h>
#define PORT 5000
#define MAXLINE 1000
// Driver code
int main()
{
        int listenfd, len;
        int value,sum=0;
        struct sockaddr_in servaddr, cliaddr;
        bzero(&servaddr, sizeof(servaddr));

        // Create a UDP Socket
        listenfd = socket(AF_INET, SOCK_DGRAM, 0);
        servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
        servaddr.sin_port = htons(PORT);
        servaddr.sin_family = AF_INET;

        // bind server address to socket descriptor
        bind(listenfd, (struct sockaddr*)&servaddr, sizeof(servaddr));
        //receive the datagram

        len = sizeof(cliaddr);

        //receive message from server

        recvfrom(listenfd, &value, sizeof(value),0, (struct sockaddr*)&cliaddr,&len);

        sum=value*(value+1);        //calculate sum

        printf("Result calculated \n");

        printf("Sending result to client...\n");

        // send the response

        sendto(listenfd, &sum, MAXLINE, 0,

                (struct sockaddr*)&cliaddr, sizeof(cliaddr));

}
```

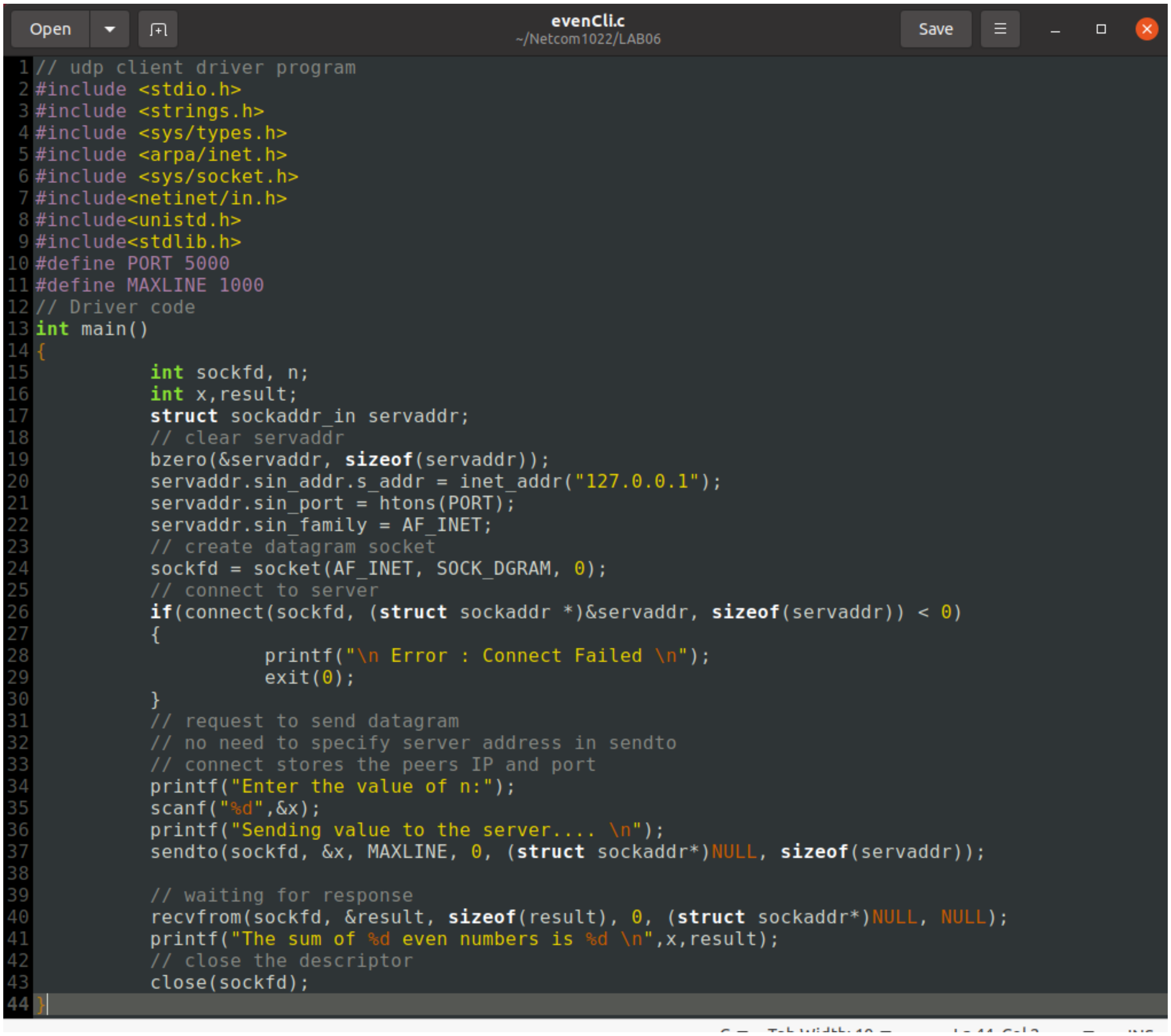
Output:



```
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$ gedit evenSer.c
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$ gcc evenSer.c
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$ ./a.out
Result calculated
Sending result to client...
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$
```

Client Program Source Code:

Code window:



```
// udp client driver 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
// Driver code
int main()
{
        int sockfd, n;
        int x,result;
        struct sockaddr_in servaddr;
        // clear servaddr
        bzero(&servaddr, sizeof(servaddr));
        servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
        servaddr.sin_port = htons(PORT);
        servaddr.sin_family = AF_INET;
        // create datagram socket
        sockfd = socket(AF_INET, SOCK_DGRAM, 0);
        // connect to server
        if(connect(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0)
        {
                printf("\n Error : Connect Failed \n");
                exit(0);
        }
        // request to send datagram
        // no need to specify server address in sendto
        // connect stores the peers IP and port
        printf("Enter the value of n:");
        scanf("%d",&x);
        printf("Sending value to the server.... \n");
        sendto(sockfd, &x, MAXLINE, 0, (struct sockaddr*)NULL, sizeof(servaddr));

        // waiting for response
        recvfrom(sockfd, &result, sizeof(result), 0, (struct sockaddr*)NULL, NULL);
        printf("The sum of %d even numbers is %d \n",x,result);
        // close the descriptor
        close(sockfd);
}
```

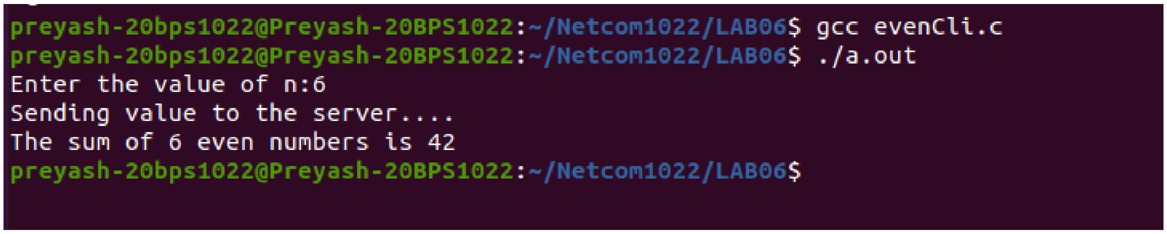
Code:

```
// udp client driver 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
// Driver code
int main()
{
    int sockfd, n;
    int x,result;
    struct sockaddr_in servaddr;
    // clear servaddr
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
    servaddr.sin_port = htons(PORT);
    servaddr.sin_family = AF_INET;
    // create datagram socket
    sockfd = socket(AF_INET, SOCK_DGRAM, 0);
    // connect to server
    if(connect(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0)
    {
        printf("\n Error : Connect Failed \n");
        exit(0);
    }
    // request to send datagram
    // no need to specify server address in sendto
```

```
        // connect stores the peers IP and port
        printf("Enter the value of n:");
        scanf("%d",&x);
        printf("Sending value to the server.... \n");
        sendto(sockfd, &x, MAXLINE, 0, (struct sockaddr*)NULL, sizeof(servaddr));

        // waiting for response
        recvfrom(sockfd, &result, sizeof(result), 0, (struct sockaddr*)NULL, NULL);
        printf("The sum of %d even numbers is %d \n",x,result);
        // close the descriptor
        close(sockfd);
}
```

Output:



```
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$ gcc evenCli.c
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$ ./a.out
Enter the value of n:6
Sending value to the server....
The sum of 6 even numbers is 42
preyash-20bps1022@Preyash-20BPS1022:~/Netcom1022/LAB06$
```

Result: We successfully created a program to perform the required output.