Name: Muyu Tong
Email: muyut@andrew.cmu.edu

Project 2 Task 0

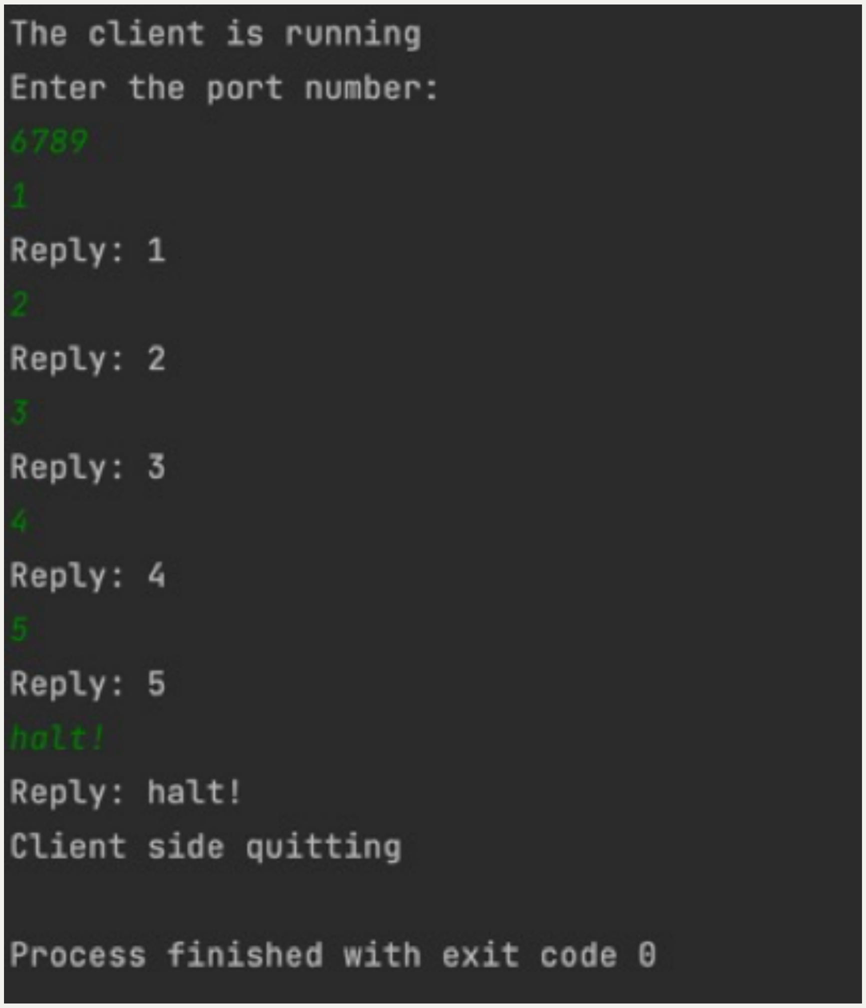
Project2Task0Client

```java
//source https://github.com/CMU-Heinz-95702/Project-2-Client-Server
public class EchoClientUDP{
    public static void main(String args[]){
        // args give message contents and server hostname
        System.out.println("The client is running");
        DatagramSocket aSocket = null;
        try {
            InetAddress aHost = InetAddress.getByName("localhost");

            //prompt the port number
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the port number: ");
            int serverPort = scanner.nextInt();

            aSocket = new DatagramSocket();
            String nextLine;
            BufferedReader typed = new BufferedReader(new InputStreamReader(System.in));
            while ((nextLine = typed.readLine()) != null) {
                //each loop read from the console and send it to the server
                byte [] m = nextLine.getBytes();
                DatagramPacket request = new DatagramPacket(m, m.length, aHost, serverPort);
                aSocket.send(request);
                byte[] buffer = new byte[1000];
                DatagramPacket reply = new DatagramPacket(buffer, buffer.length);
                aSocket.receive(reply);
                byte[] message = new byte[reply.getLength()];
                System.arraycopy(reply.getData(), 0, message, 0, message.length);
                System.out.println("Reply: " + new String(message));
                if (new String(message).equals("halt!")) {
                    System.out.println("Client side quitting");
                    break;
                }
            }

        }catch (SocketException e) {System.out.println("Socket: " + e.getMessage());
        }catch (IOException e){System.out.println("IO: " + e.getMessage());
        }finally {if(aSocket != null) aSocket.close();}
    }
}

```

Project2Task0Server

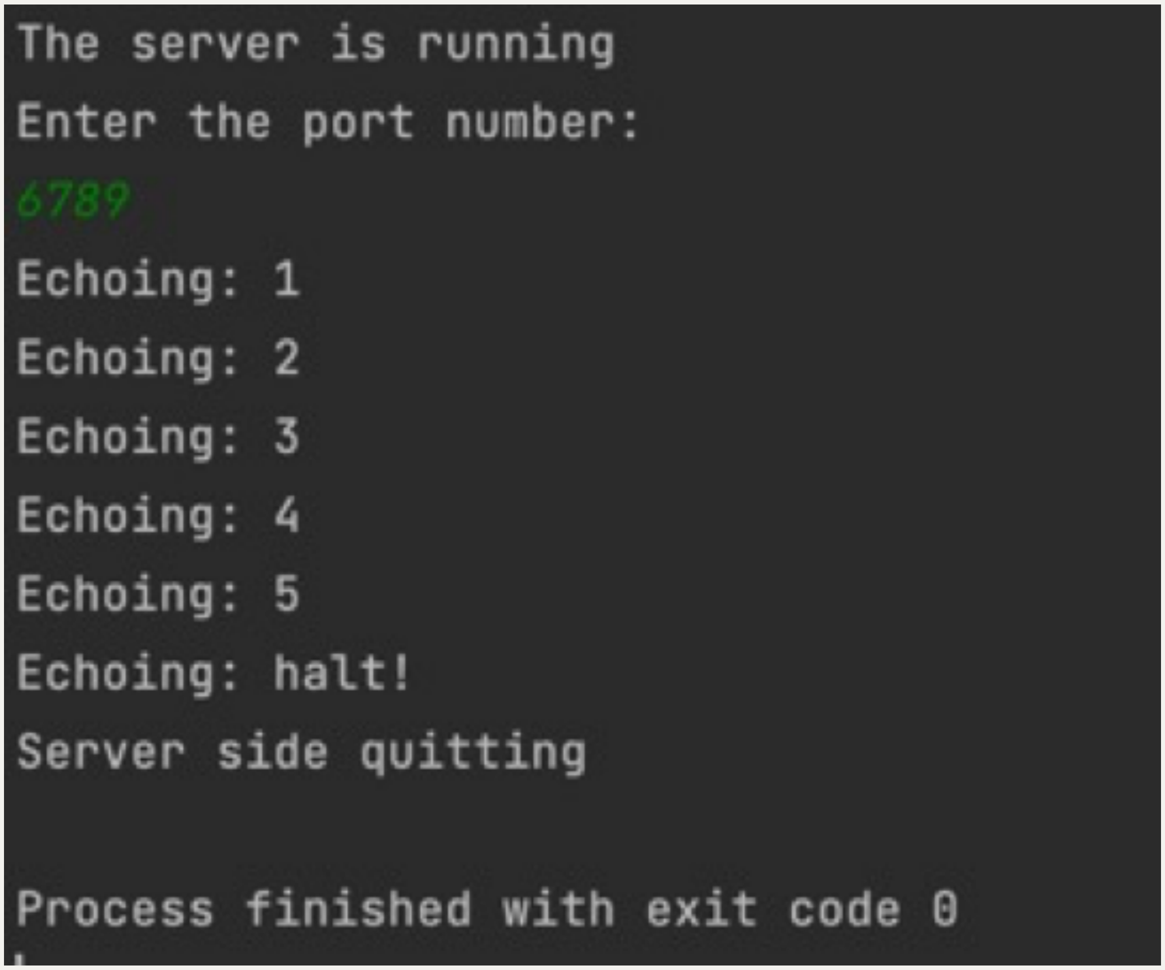
```java
//source https://github.com/CMU-Heinz-95702/Project-2-Client-Server
public class EchoServerUDP{
    public static void main(String args[]){
        System.out.println("The server is running");
        DatagramSocket aSocket = null;
        byte[] buffer = new byte[1000];
        try{
            //prompt the port number
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the port number: ");
            int serverPort = scanner.nextInt();
            scanner.close();

            aSocket = new DatagramSocket(serverPort);
            DatagramPacket request = new DatagramPacket(buffer, buffer.length);
            while(true){
                //each round receive a request from the client
                aSocket.receive(request);
                DatagramPacket reply = new DatagramPacket(request.getData(),
                        request.getLength(), request.getAddress(), request.getPort());
                byte[] message = new byte[request.getLength()];
                System.arraycopy(request.getData(), 0, message, 0, message.length);
                String requestString = new String(message);
                System.out.println("Echoing: "+requestString);
                aSocket.send(reply);
                if (requestString.equals("halt!")) {
                    System.out.println("Server side quitting");
                    break;
                }
            }
        }catch (SocketException e){System.out.println("Socket: " + e.getMessage());
        }catch (IOException e) {System.out.println("IO: " + e.getMessage());
        }finally {if(aSocket != null) aSocket.close();}
    }
}
```

Project2Task0ClientConsole



```
The client is running
Enter the port number:
6789
1
Reply: 1
2
Reply: 2
3
Reply: 3
4
Reply: 4
5
Reply: 5
halt!
Reply: halt!
Client side quitting

Process finished with exit code 0
```

Project2Task0ServerConsole



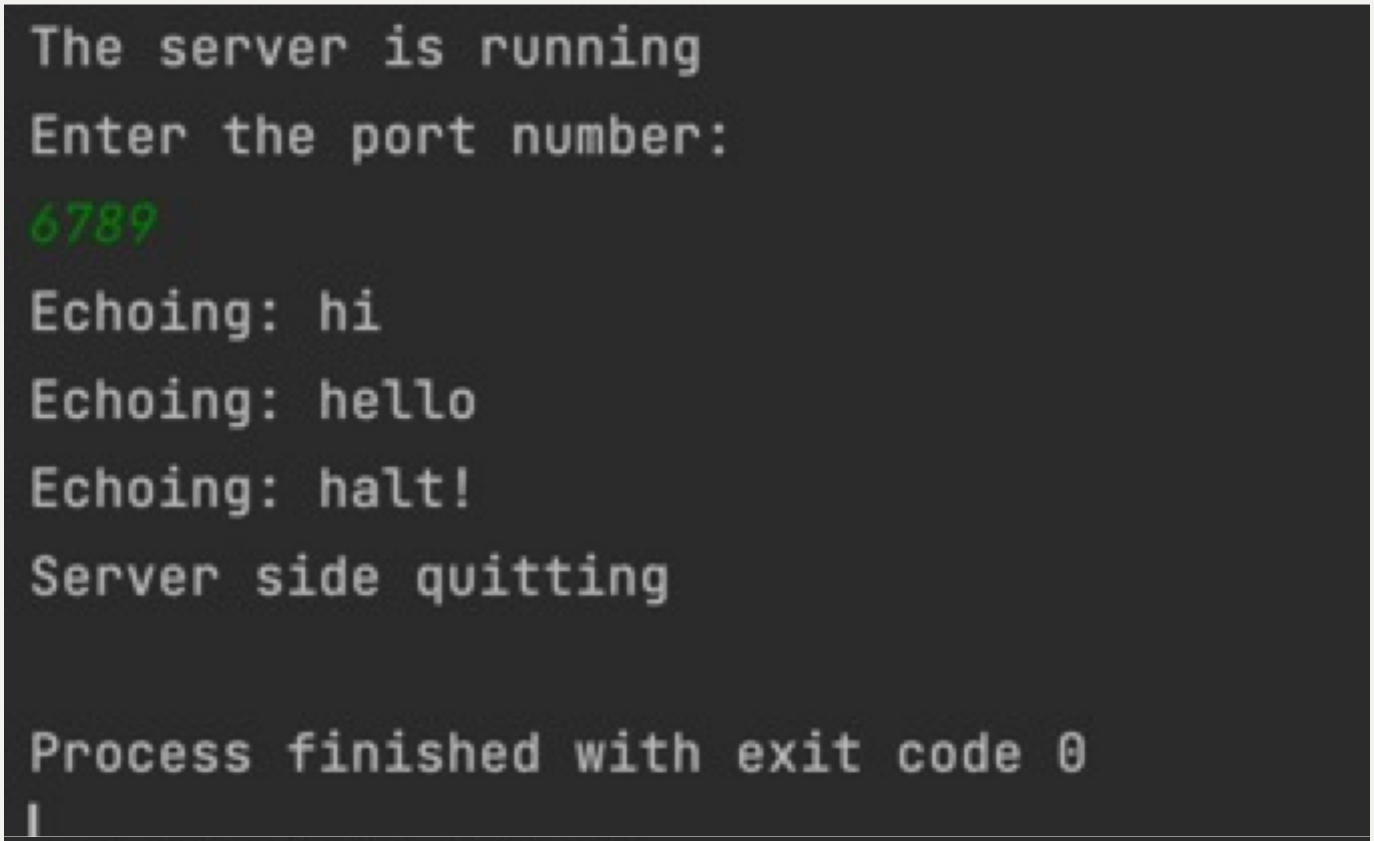
```
The server is running
Enter the port number:
6789
Echoing: 1
Echoing: 2
Echoing: 3
Echoing: 4
Echoing: 5
Echoing: halt!
Server side quitting

Process finished with exit code 0
```

Project 2 Task 1

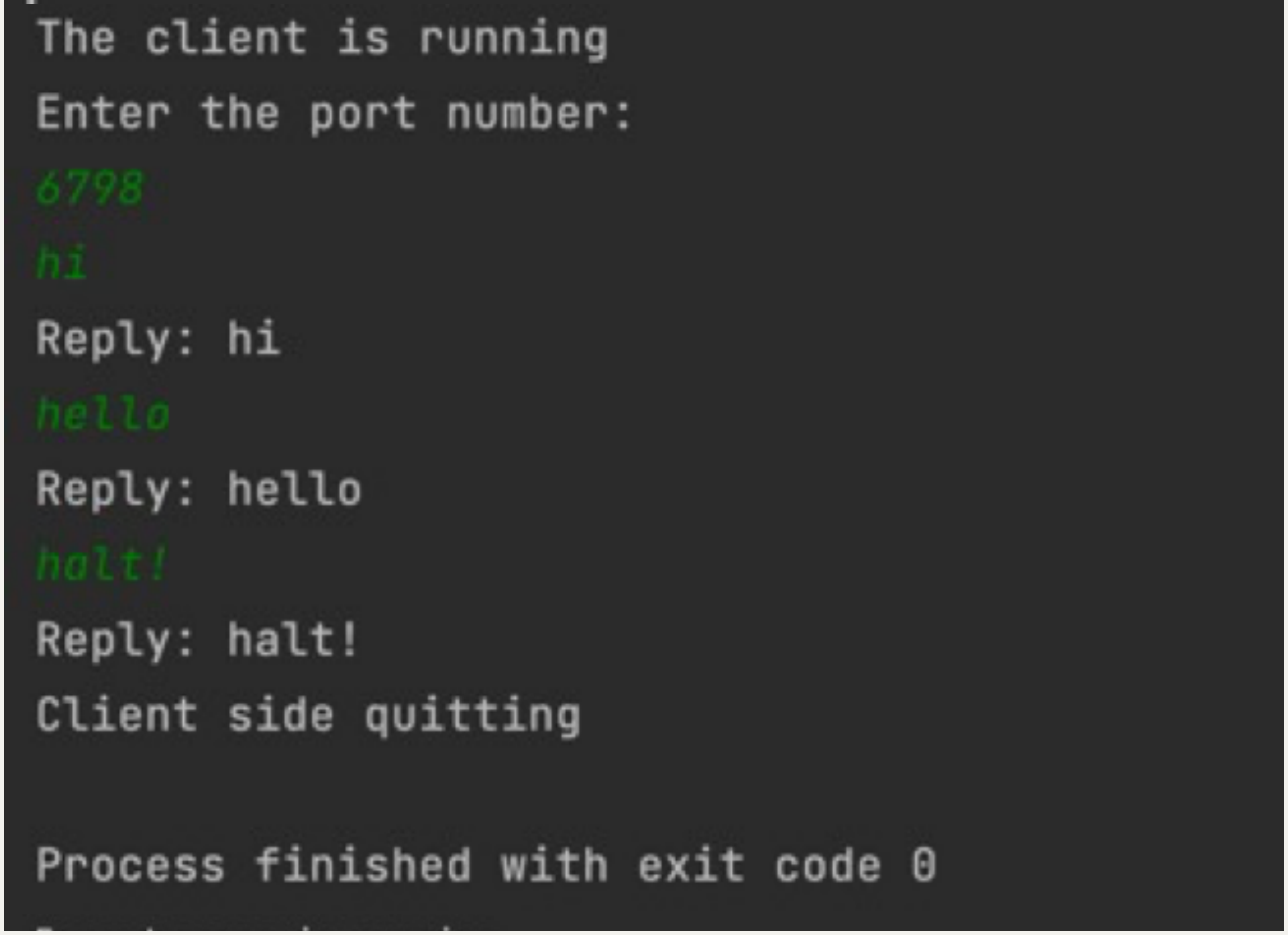
EavesdropperUDP.java

```java
//source https://github.com/CMU-Heinz-95702/Project-2-Client-Server
public class EavesdropperUDP {
    DatagramSocket aSocket;
    DatagramSocket bSocket;
    int masPort;

    public EavesdropperUDP() throws SocketException {
        System.out.println("Eavesdropper is running");
        //prompt the port number
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the masquerading port number: ");
        masPort = scanner.nextInt(); scanner.nextLine();
        System.out.println("Enter the port number: ");
        int portNum = scanner.nextInt(); scanner.nextLine();
        scanner.close();

        aSocket = new DatagramSocket(portNum);  //socket listen to 6798
        bSocket = new DatagramSocket();

        byte[] buffer = new byte[1000];
        while (true) {
            //get request from client
            DatagramPacket request = new DatagramPacket(buffer, buffer.length);
            try {
                aSocket.receive(request);
                byte[] message = new byte[request.getLength()];
                System.arraycopy(request.getData(), 0, message, 0, message.length);
                System.out.println("Eavesdrop from client: " + new String(message));
            } catch (IOException e) {
                e.printStackTrace();
            }

            //transmit the data to server
            DatagramPacket transmit = new DatagramPacket(request.getData(),
                    request.getLength(), request.getAddress(), masPort);
            try {
                bSocket.send(transmit);
            } catch (IOException ioException) {
                ioException.printStackTrace();
            }

            //receive reply from the server and sent it back to the client
            DatagramPacket reply = new DatagramPacket(buffer, buffer.length);
            try {
                bSocket.receive(reply);
                byte[] message = new byte[reply.getLength()];
                System.arraycopy(reply.getData(), 0, message, 0, message.length);
                System.out.println("Eavesdrop from server: " + new String(message));
                reply.setPort(request.getPort());
                aSocket.send(reply);
            } catch (IOException ioException) {
                ioException.printStackTrace();
            }
        }
    }

    public static void main(String[] args) {
        try {
            new EavesdropperUDP();
        } catch (SocketException e) {
            e.printStackTrace();
        }
    }

}
```

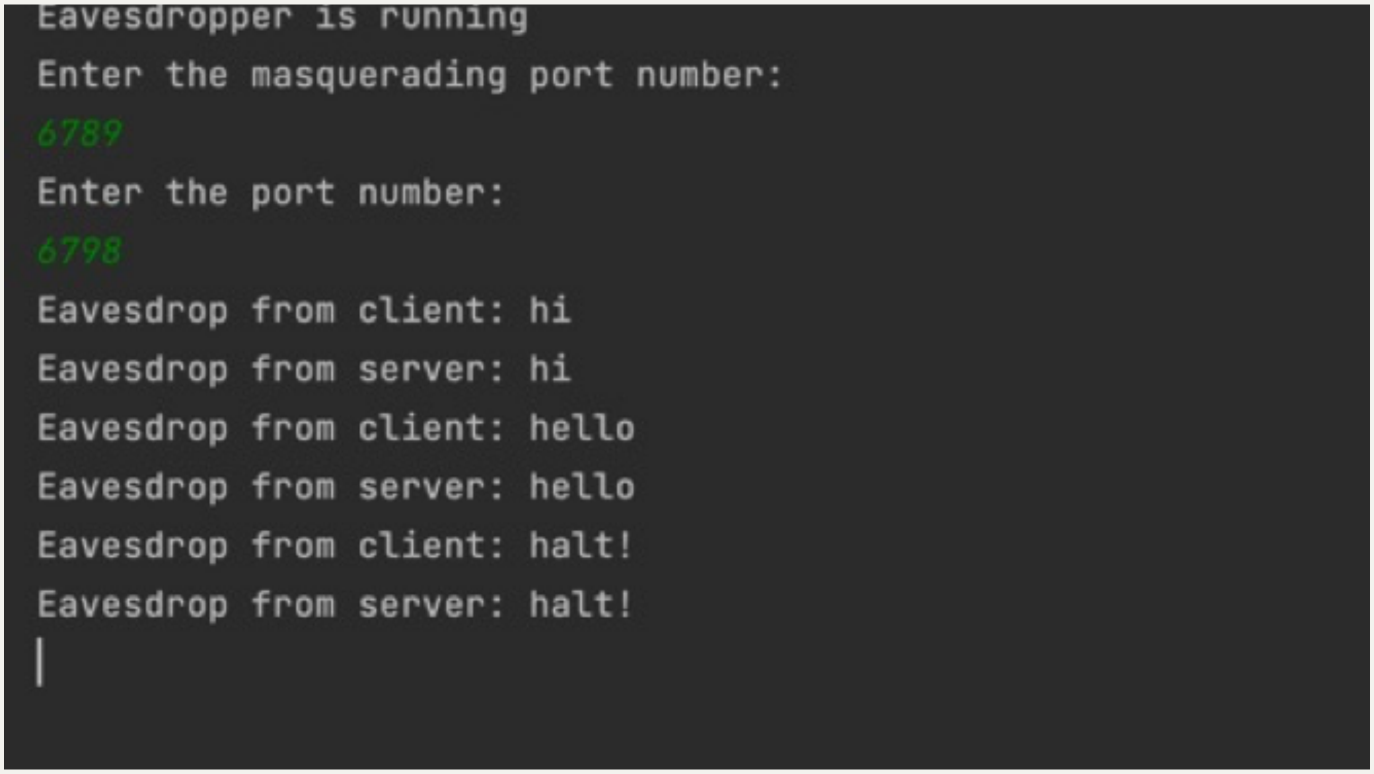
Project2Task1ThreeConsoles



```
The server is running
Enter the port number:
6789
Echoing: hi
Echoing: hello
Echoing: halt!
Server side quitting


Process finished with exit code 0
```



```
The client is running
Enter the port number:
6798
hi
Reply: hi
hello
Reply: hello
halt!
Reply: halt!
Client side quitting


Process finished with exit code 0
```



```
Eavesdropper is running
Enter the masquerading port number:
6789
Enter the port number:
6798
Eavesdrop from client: hi
Eavesdrop from server: hi
Eavesdrop from client: hello
Eavesdrop from server: hello
Eavesdrop from client: halt!
Eavesdrop from server: halt!
```

Project 2 Task 2

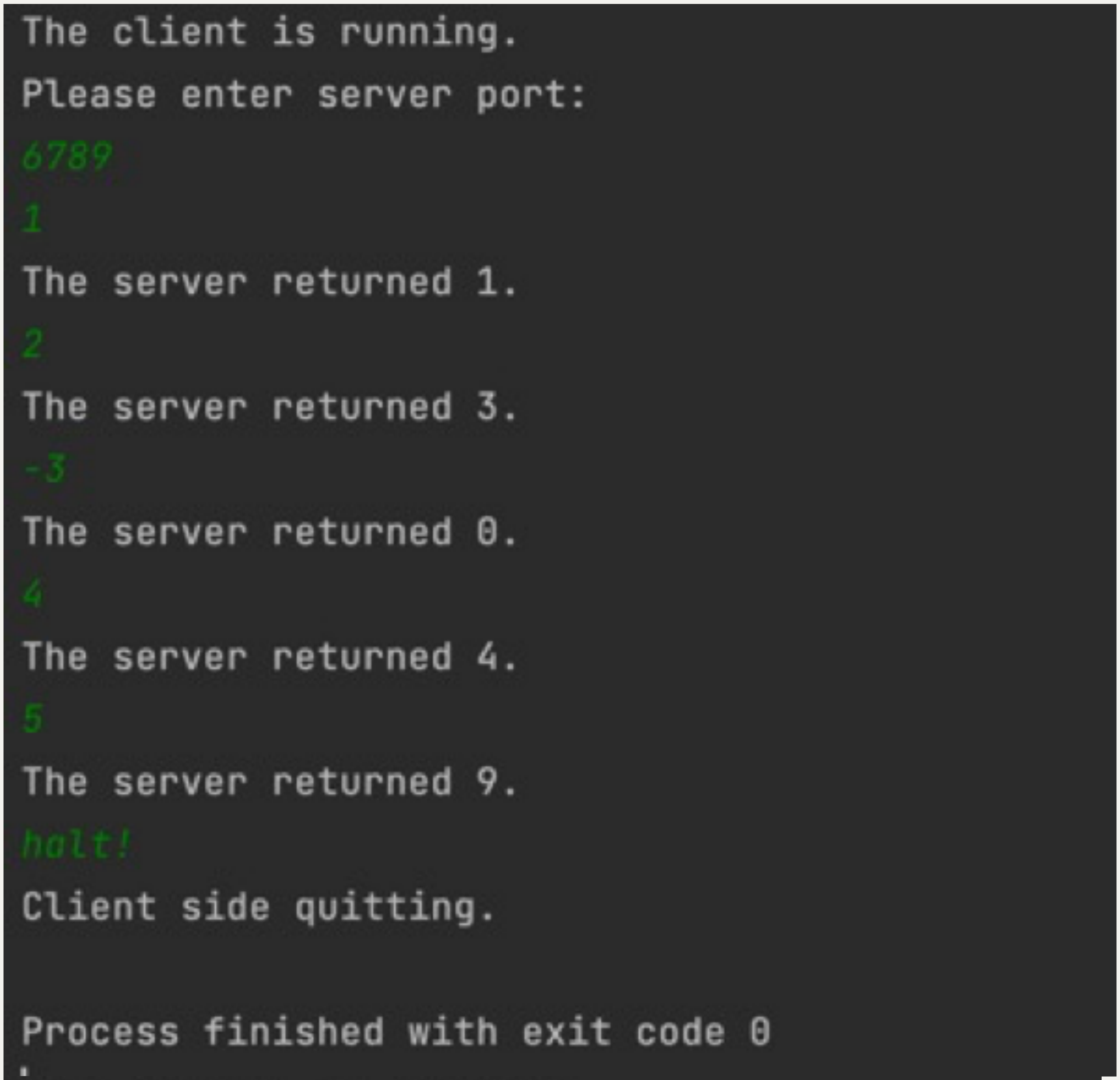
Project2Task2Client

```java
//source https://github.com/CMU-Heinz-95702/Project-2-Client-Server
public class AddingClientUDP {
    public static DatagramSocket socket;
    public static int portNum;
    public static void main(String[] args) {
        System.out.println("The client is running.");
        try {
            socket = new DatagramSocket();
            System.out.println("Please enter server port: ");
            Scanner scanner = new Scanner(System.in);
            portNum = scanner.nextInt();scanner.nextLine();

            String s;
            while (scanner.hasNextLine()) {
                s = scanner.nextLine();
                if (s.equals("halt!")) {
                    System.out.println("Client side quitting.");
                    break;
                }
                int num = Integer.parseInt(s);
                int res = add(num);
                System.out.println("The server returned " + res + ".");
            }
        } catch (SocketException e) {
            e.printStackTrace();
        }
    }
    public static int add(int i) {
        String num = String.valueOf(i);
        byte[] m = num.getBytes();
        byte[] buffer = new byte[1000];
        try {
            DatagramPacket request = new DatagramPacket(m, m.length,InetAddress.getByName("localhost"),portNum);
            socket.send(request);
            DatagramPacket reply = new DatagramPacket(buffer, buffer.length);
            socket.receive(reply);
            byte[] message = new byte[reply.getLength()];
            System.arraycopy(reply.getData(), 0, message, 0, reply.getLength());
            String res = new String(message);
            return Integer.parseInt(res);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return 0;
    }
}
```

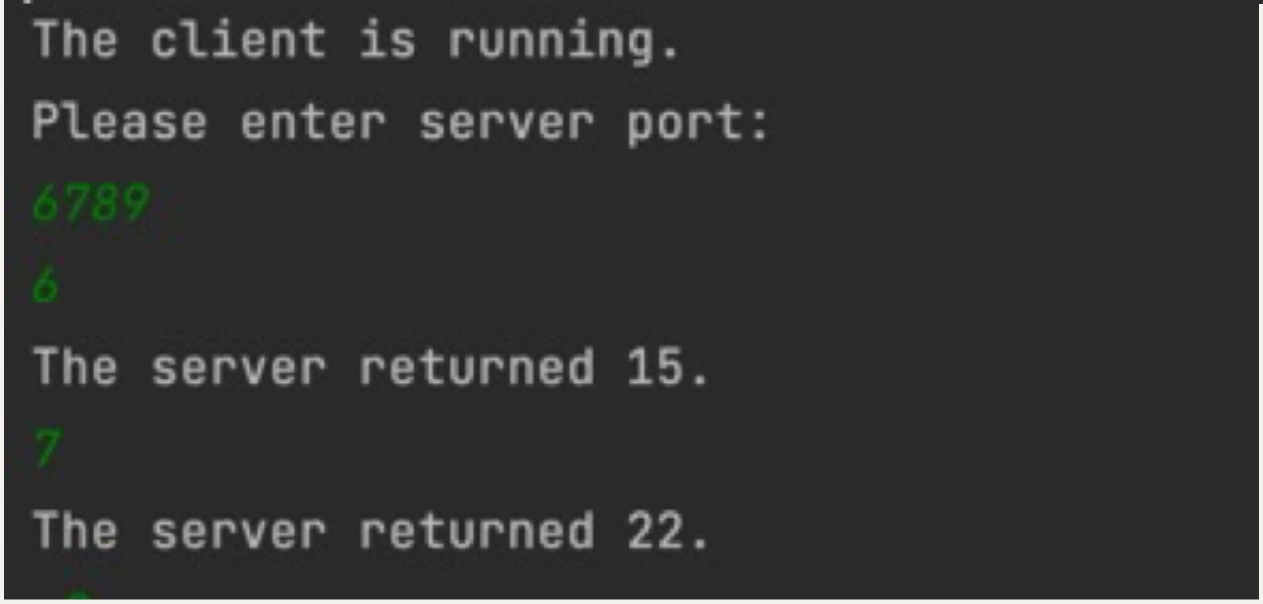
Project2Task2Server

```java
//source https://github.com/CMU-Heinz-95702/Project-2-Client-Server
public class AddingServerUDP {
    public static int sum = 0;
    public static void main(String[] args) {
        DatagramSocket socket = null;
        try {
            socket = new DatagramSocket(6789);
        } catch (SocketException e) {
            e.printStackTrace();
        }
        while (true) {
            byte[] buffer = new byte[1000];
            DatagramPacket request = new DatagramPacket(buffer, buffer.length);
            try {
                socket.receive(request);
            } catch (IOException e) {
                e.printStackTrace();
            }

            byte[] message = new byte[request.getLength()];
            System.arraycopy(request.getData(), 0, message, 0, request.getLength());
            int num = Integer.parseInt(new String(message));
            System.out.printf("Adding %d to %d\n", num, sum);
            add(num);
            System.out.printf("Returning sum of %d to client\n", sum);
            byte[] res = String.valueOf(sum).getBytes();
            DatagramPacket reply = new DatagramPacket(res, res.length, request.getAddress(), request.getPort());
            try {
                socket.send(reply);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    public static void add(int i) {
        sum += i;
    }
}
```

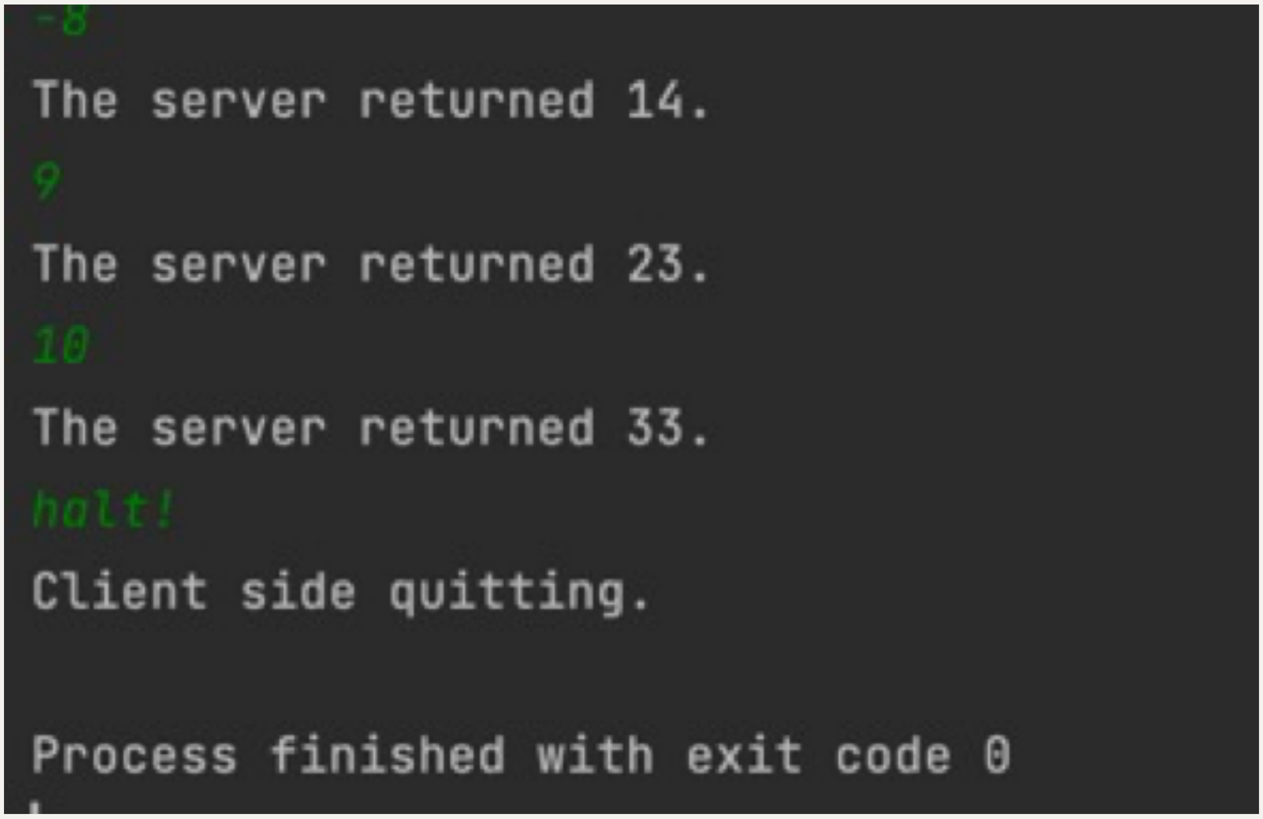
Project2Task2ClientConsole



```
The client is running.
Please enter server port:
6789
1
The server returned 1.
2
The server returned 3.
-3
The server returned 0.
4
The server returned 4.
5
The server returned 9.
halt!
Client side quitting.

Process finished with exit code 0
```

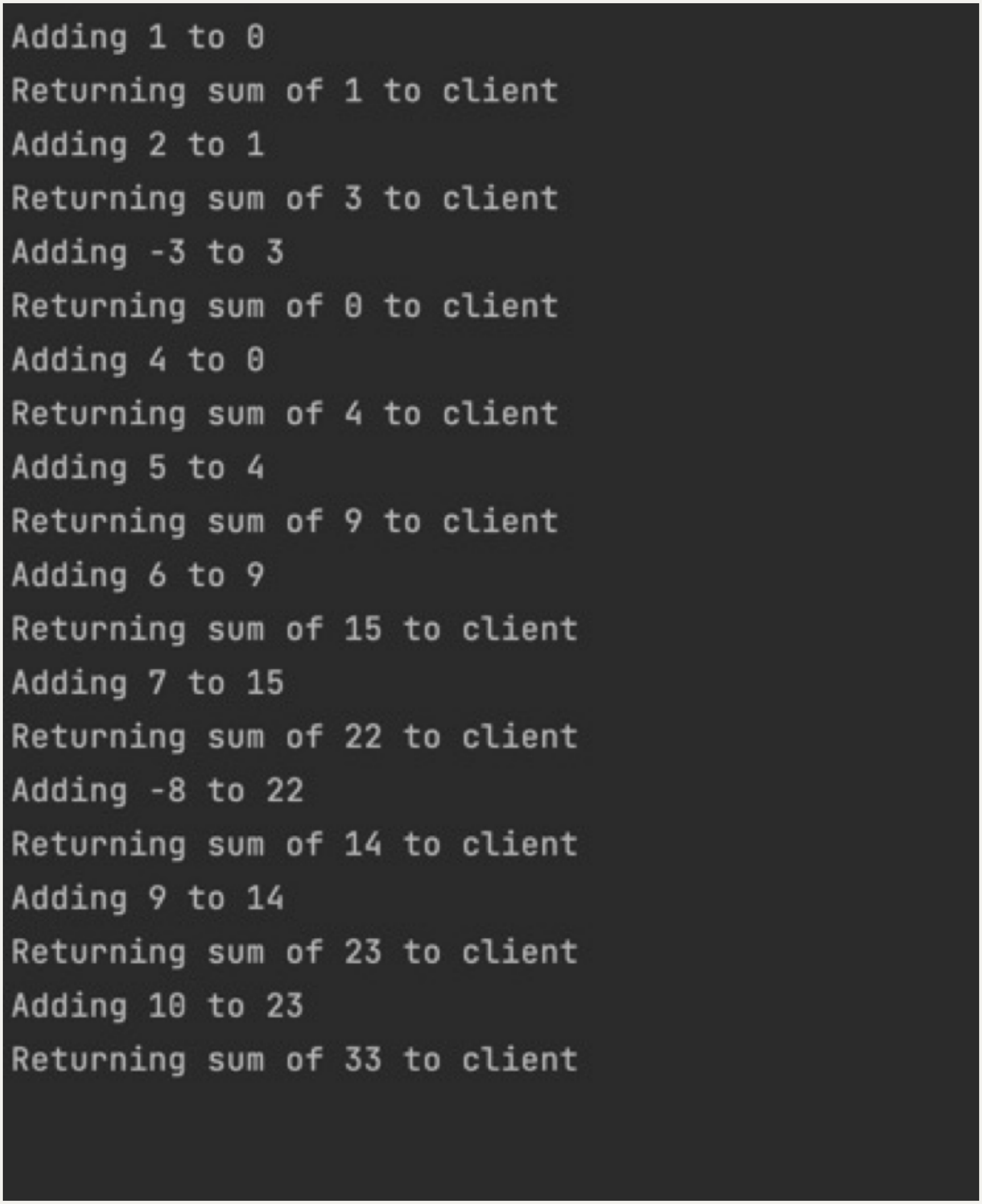


```
The client is running.
Please enter server port:
6789
6
The server returned 15.
7
The server returned 22.
```



```
-8
The server returned 14.
9
The server returned 23.
10
The server returned 33.
halt!
Client side quitting.


Process finished with exit code 0
```

Project2Task2ServerConsole



```
Adding 1 to 0
Returning sum of 1 to client
Adding 2 to 1
Returning sum of 3 to client
Adding -3 to 3
Returning sum of 0 to client
Adding 4 to 0
Returning sum of 4 to client
Adding 5 to 4
Returning sum of 9 to client
Adding 6 to 9
Returning sum of 15 to client
Adding 7 to 15
Returning sum of 22 to client
Adding -8 to 22
Returning sum of 14 to client
Adding 9 to 14
Returning sum of 23 to client
Adding 10 to 23
Returning sum of 33 to client
```

Project 2 Task 3

Project2Task3Client

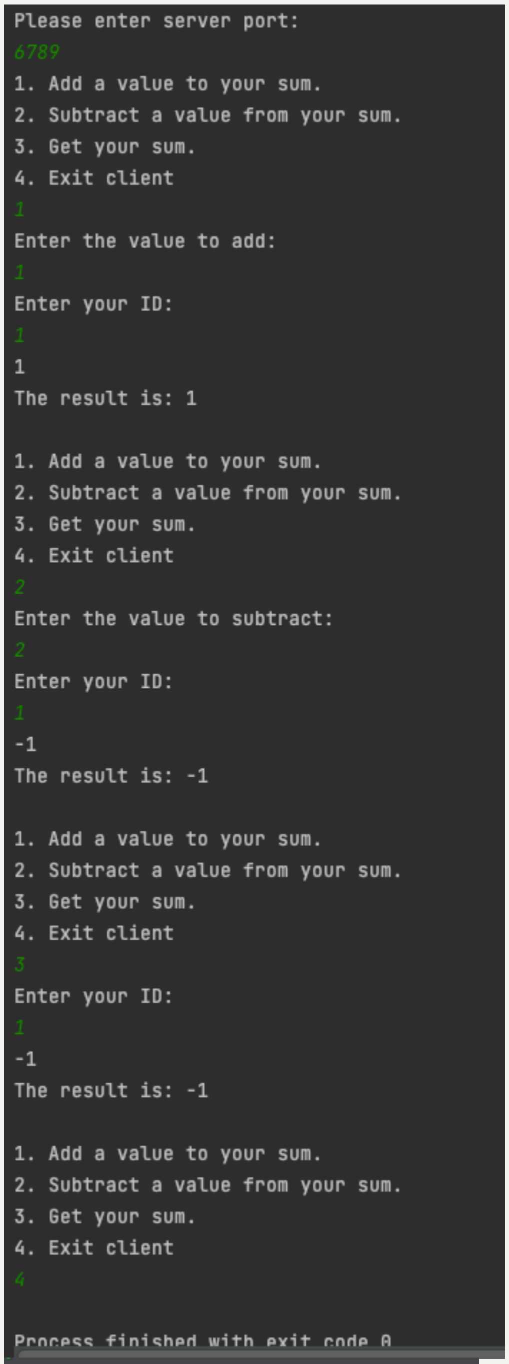
```java
//source https://github.com/CMU-Heinz-95702/Project-2-Client-Server
public class RemoteVariableClientUDP {
    public static DatagramSocket socket;
    public static int portNum;

    public RemoteVariableClientUDP() {
    }

    public static void main(String[] args) {
        System.out.println("The client is running.");

        try {
            socket = new DatagramSocket();
            System.out.println("Please enter server port: ");
            Scanner scanner = new Scanner(System.in);
            portNum = scanner.nextInt();
            scanner.nextLine();

            while(true) {
                System.out.println("1. Add a value to your sum.\n2. Subtract a value from your sum.\n3. Get your sum.\n4. Exit client");
                String s = scanner.nextLine();
                int choice = Integer.parseInt(s);
                int num;
                int res;
                int id;
                if (choice == 1) {
                    System.out.println("Enter the value to add: ");
                    num = Integer.parseInt(scanner.nextLine());
                    System.out.println("Enter your ID: ");
                    id = Integer.parseInt(scanner.nextLine());
                    res = add(num, id);
                } else if (choice == 2) {
                    System.out.println("Enter the value to subtract:");
                    num = Integer.parseInt(scanner.nextLine());
                    System.out.println("Enter your ID: ");
                    id = Integer.parseInt(scanner.nextLine());
                    res = subtract(num, id);
                } else {
                    if (choice != 3) {
                        return;
                    }

                    System.out.println("Enter your ID: ");
                    id = Integer.parseInt(scanner.nextLine());
                    res = get(id);
                }

                System.out.println("The result is: " + res);
                System.out.println();
            }
        } catch (SocketException var7) {
            var7.printStackTrace();
        }
    }

    public static int add(int i, int id) {
        String request = "\"add\"," + i + "," + id;
        return request(request);
    }

    public static int subtract(int i, int id) {
        String request = "\"subtract\"," + i + "," + id;
        return request(request);
    }

    public static int get(int id) {
        String request = "\"get\",0," + id;
        return request(request);
    }

    public static int request(String requestBody) {
        byte[] m = requestBody.getBytes();
        byte[] buffer = new byte[1000];

        try {
            DatagramPacket request = new DatagramPacket(m,
m.length, InetAddress.getByName("localhost"), portNum);
            socket.send(request);
            DatagramPacket reply = new DatagramPacket(buffer,
buffer.length);
            socket.receive(reply);
            byte[] message = new byte[reply.getLength()];
            System.arraycopy(reply.getData(), 0, message, 0,
reply.getLength());
            String res = new String(message);
            System.out.println(res);
            return Integer.parseInt(res);
        } catch (IOException var7) {
            var7.printStackTrace();
            return 0;
        }
    }
}

```

Project2Task3Server

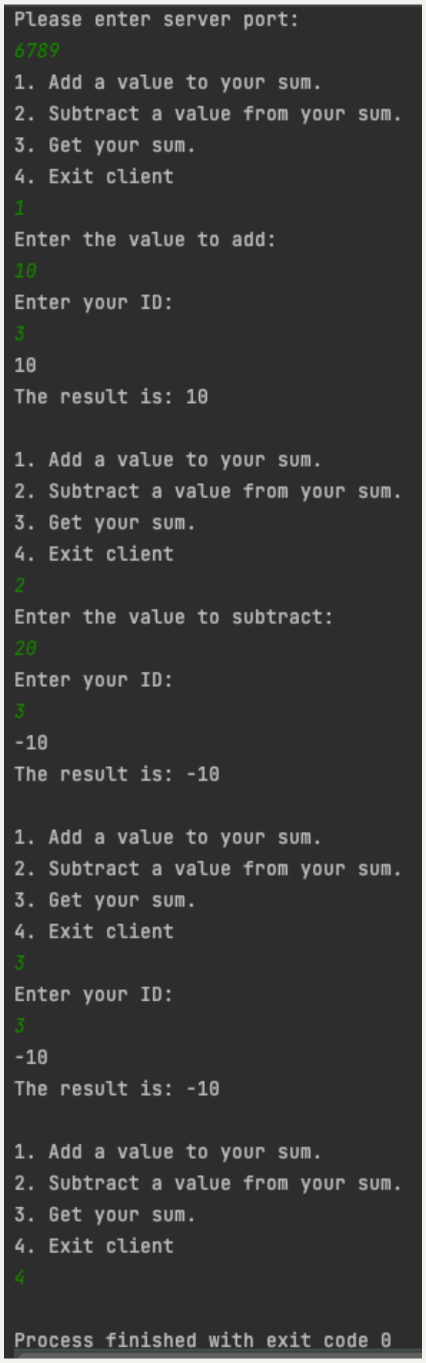
```java
//source https://github.com/CMU-Heinz-95702/Project-2-Client-Server
public class RemoteVariableServerUDP {
    public static TreeMap<Integer, Integer> map = new TreeMap<>();
    public static void main(String[] args) {
        DatagramSocket socket = null;
        try {
            socket = new DatagramSocket(6789);
        } catch (SocketException e) {
            e.printStackTrace();
        }
        while (true) {
            byte[] buffer = new byte[1000];
            DatagramPacket request = new DatagramPacket(buffer, buffer.length);
            try {
                socket.receive(request);
            } catch (IOException e) {
                e.printStackTrace();
            }

            byte[] message = new byte[request.getLength()];
            System.arraycopy(request.getData(), 0, message, 0, request.getLength());
            String requestBody = new String(message);
            String[] params = requestBody.split(",");
            String method = params[0];
            int num = Integer.parseInt(params[1]);
            int id = Integer.parseInt(params[2]);

            int sum = exec(method, id, num);

            byte[] res = String.valueOf(sum).getBytes();
            DatagramPacket reply = new DatagramPacket(res, res.length, request.getAddress(), request.getPort());
            try {
                socket.send(reply);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    public static int exec(String method, int id, int num) {
        if (!map.containsKey(id)) map.put(id, 0);
        int res;
        if (method.equals("get")) {
            res = get(id);
        } else if (method.equals("add")) {
            res = add(id, num);
        } else {
            res = subtract(id, num);
        }
        System.out.printf("Visitor id: %d, method: %s, returned value: %d\n", id, method, res);
        return map.get(id);
    }

    public static int add(int id, int num) {
        map.put(id,map.get(id)+num);
        return map.get(id);
    }

    public static int subtract(int id, int num) {
        map.put(id,map.get(id)-num);
        return map.get(id);
    }

    public static int get(int id) {
        return map.get(id);
    }
}
```

```
68
```

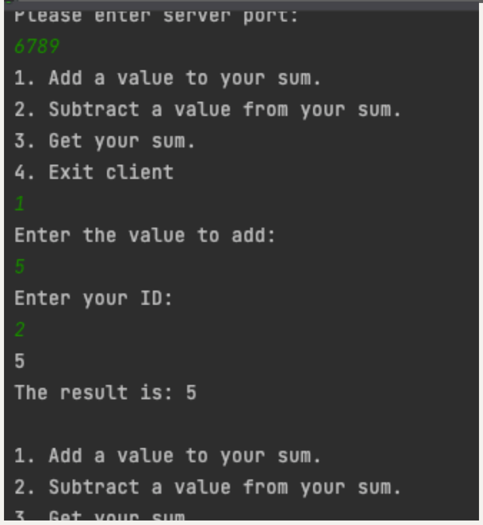
Project2Task3ClientConsole



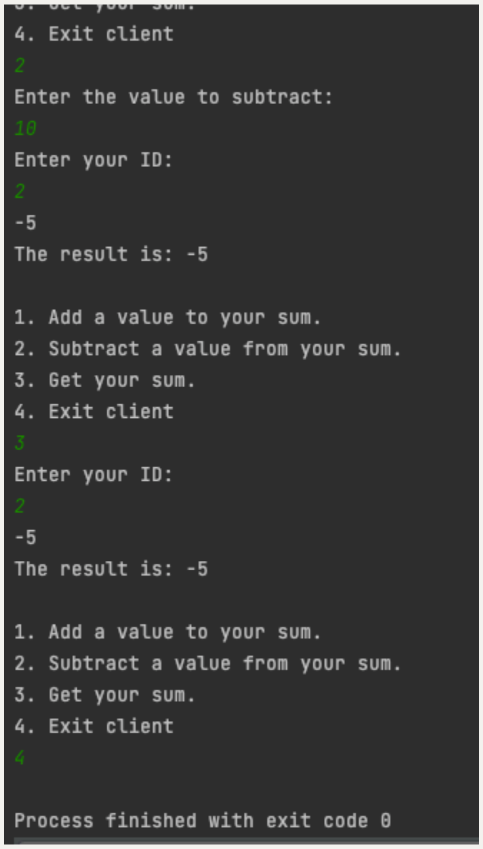
```
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
1
Enter the value to add:
1
Enter your ID:
1
1
The result is: 1

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
2
Enter the value to subtract:
2
Enter your ID:
1
-1
The result is: -1

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
1
-1
The result is: -1

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```



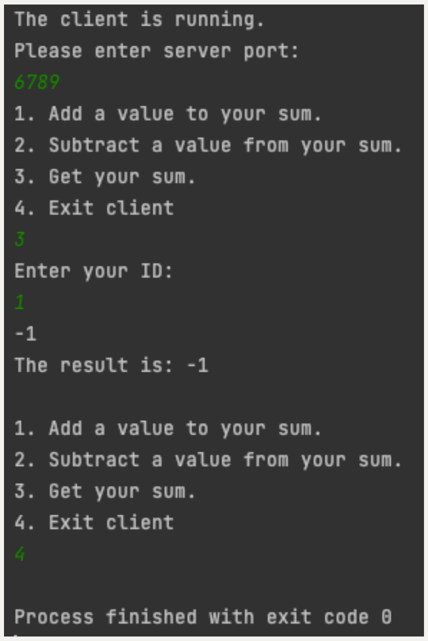
```
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
1
Enter the value to add:
10
Enter your ID:
3
10
The result is: 10

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
2
Enter the value to subtract:
20
Enter your ID:
3
-10
The result is: -10

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
3
-10
The result is: -10

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```



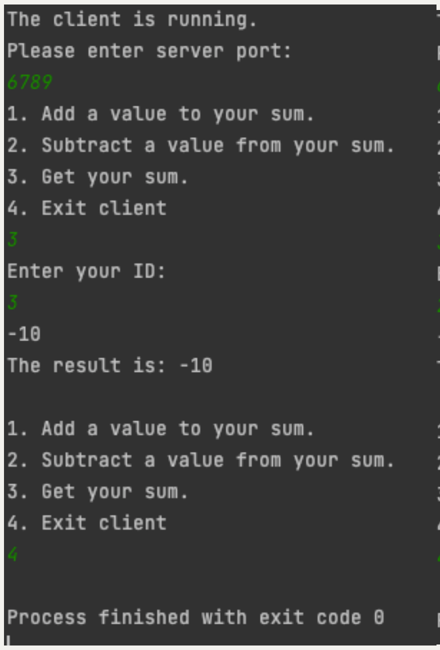
```
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
1
Enter the value to add:
5
Enter your ID:
2
5
The result is: 5

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum
```



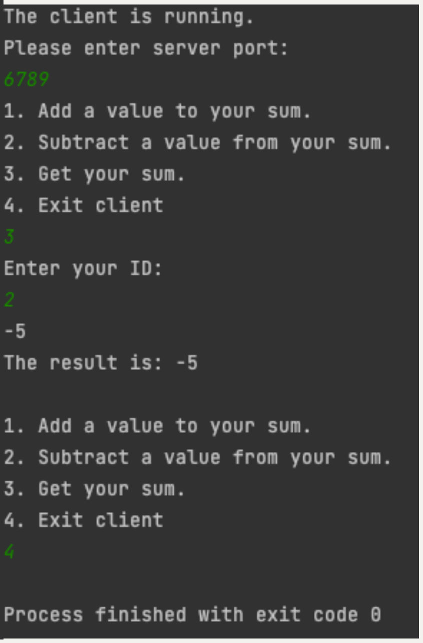
```
4. Exit client
2
Enter the value to subtract:
10
Enter your ID:
2
-5
The result is: -5

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
2
-5
The result is: -5

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```



```
The client is running.
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
1
-1
The result is: -1

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```

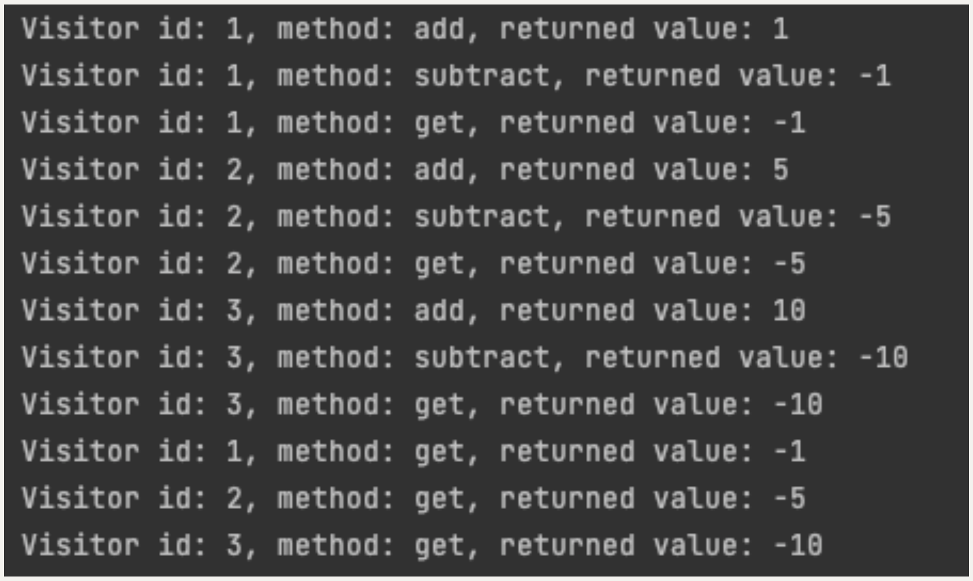


```
The client is running.
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
3
-10
The result is: -10

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```



```
The client is running.
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
2
-5
The result is: -5

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```

Project2Task3ServerConsole


```
Visitor id: 1, method: add, returned value: 1
Visitor id: 1, method: subtract, returned value: -1
Visitor id: 1, method: get, returned value: -1
Visitor id: 2, method: add, returned value: 5
Visitor id: 2, method: subtract, returned value: -5
Visitor id: 2, method: get, returned value: -5
Visitor id: 3, method: add, returned value: 10
Visitor id: 3, method: subtract, returned value: -10
Visitor id: 3, method: get, returned value: -10
Visitor id: 1, method: get, returned value: -1
Visitor id: 2, method: get, returned value: -5
Visitor id: 3, method: get, returned value: -10
```

Project 2 Task 4

Project2Task4Client

```
//source https://github.com/CMU-Heinz-95702/Project-2-Client-Server
public class RemoteVariableClientTCP {
    public static Socket socket;
    public static int portNum = 0;
    public static void main(String[] args) {
        System.out.println("The client is running.");
        System.out.println("Please enter server port: ");
        Scanner scanner = new Scanner(System.in);
        portNum = scanner.nextInt();
        scanner.nextLine();

        String s;
        while (true) {
            System.out.println("""
                1. Add a value to your sum.
                2. Subtract a value from your sum.
                3. Get your sum.
```

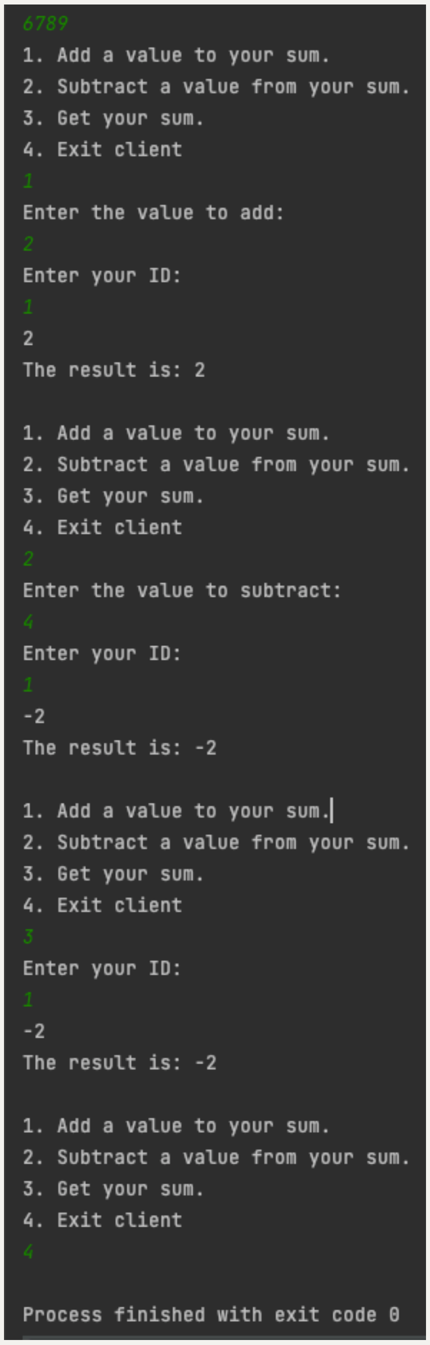
```java
                4. Exit client""");
            s = scanner.nextLine();
            int choice = Integer.parseInt(s);

            int num;
            int res;
            int id;
            if (choice == 1) {
                System.out.println("Enter the value to add: ");
                num = Integer.parseInt(scanner.nextLine());
                System.out.println("Enter your ID: ");
                id = Integer.parseInt(scanner.nextLine());
                res = add(num, id);
            } else if (choice == 2) {
                System.out.println("Enter the value to
subtract:");
                num = Integer.parseInt(scanner.nextLine());
                System.out.println("Enter your ID: ");
                id = Integer.parseInt(scanner.nextLine());
                res = subtract(num, id);
            } else if (choice == 3) {
                System.out.println("Enter your ID: ");
                id = Integer.parseInt(scanner.nextLine());
                res = get(id);
            } else {
                return;
            }

            System.out.println("The result is: " + res);
            System.out.println();
        }
    }

    public static int add(int i, int id) {
        String request = "add,"+i+","+id;
        return request(request);
    }

    public static int subtract(int i, int id) {
        String request = "subtract,"+i+","+id;
        return request(request);
    }

    public static int get(int id) {
        String request = "get,"+0+","+id;
        return request(request);
    }

    public static int request(String requestBody) {
        try {
            socket = new Socket("localhost", portNum);
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(new
BufferedWriter(new OutputStreamWriter(socket.getOutputStream())));
            out.println(requestBody);
            out.println();
            out.flush();
            String res = in.readLine();
            System.out.println(res);
            return Integer.parseInt(res);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return 0;
    }
}

```

Project2Task4Server

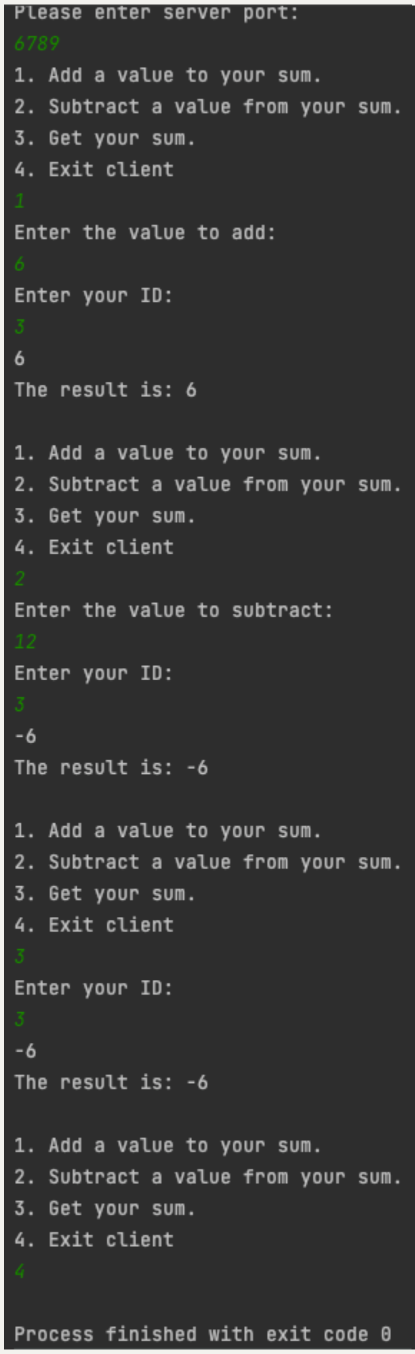
```
//source https://github.com/CMU-Heinz-95702/Project-2-Client-Server
public class RemoteVariableServerTCP {
    public static TreeMap<Integer, Integer> map = new TreeMap<>();
    public static void main(String[] args) {
        Socket socket = null;
        ServerSocket listenSocket = null;
        try {
            listenSocket = new ServerSocket(6789);
        } catch (IOException e) {
            e.printStackTrace();
        }

        while (true) {
            try {
                socket = listenSocket.accept();
                Scanner in = new Scanner(socket.getInputStream());
                StringBuilder request = new StringBuilder();
                String s;
                while (in.hasNextLine()) {
                    s = in.nextLine();
                    if (s.equals("")) {
                        String requestBody = request.toString();

                        int sum = exec(requestBody);

                        byte[] res = String.valueOf(sum).getBytes();
                        PrintWriter out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socket.getOutputStream())));
                        out.println(new String(res));
                        out.flush();
                        socket.close();
                    } else {
                        System.out.println(s);
                        request.append(s);
```

```java
                    }
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    public static int exec(String requestBody) {
        String[] params = requestBody.split(",");
        String method = params[0];
        int num = Integer.parseInt(params[1]);
        int id = Integer.parseInt(params[2]);
        if (!map.containsKey(id)) map.put(id, 0);
        int res;
        if (method.equals("get")) {
            res = get(id);
        } else if (method.equals("add")) {
            res = add(id, num);
        } else {
            res = subtract(id, num);
        }
        System.out.printf("Visitor id: %d, method: %s, returned value: %d\n", id, method, res);
        return map.get(id);
    }

    public static int add(int id, int num) {
        map.put(id,map.get(id)+num);
        return map.get(id);
    }

    public static int subtract(int id, int num) {
        map.put(id,map.get(id)-num);
        return map.get(id);
    }

    public static int get(int id) {
        return map.get(id);
    }
}

```

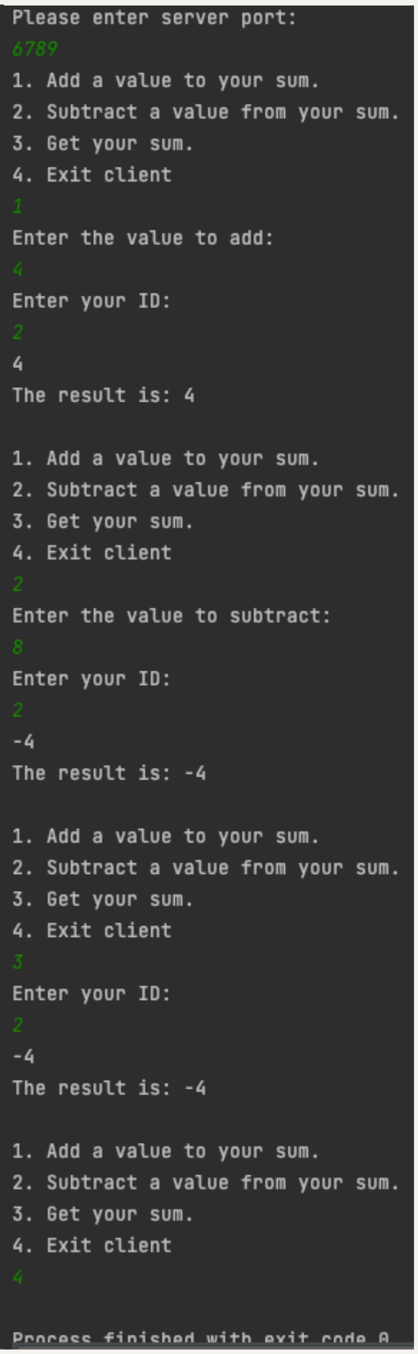
Project2Task4ClientConsole



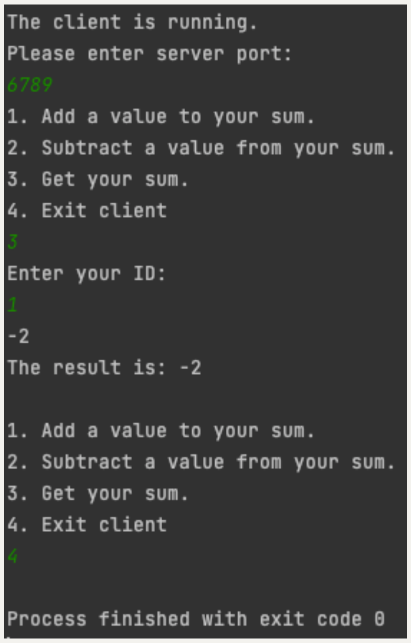
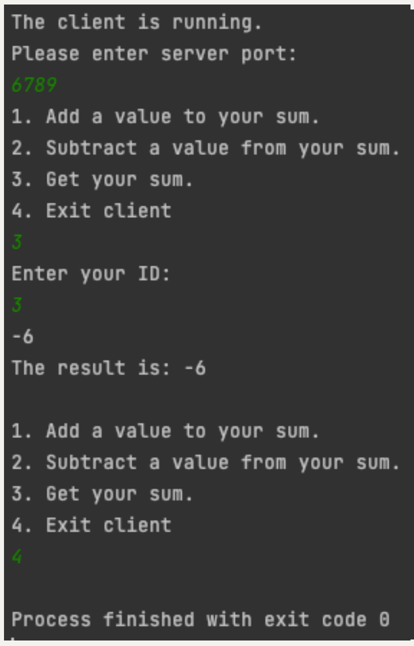
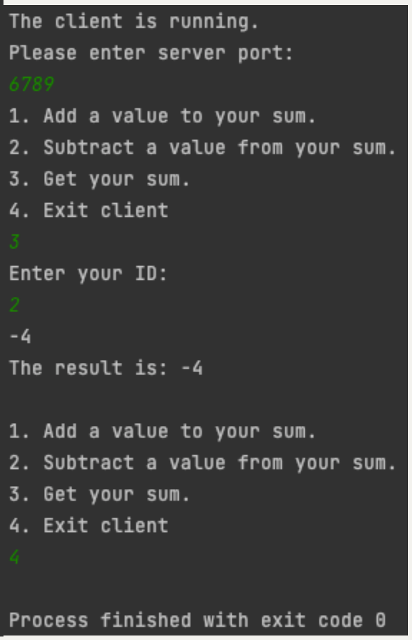
```
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
1
Enter the value to add:
2
Enter your ID:
1
2
The result is: 2

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
2
Enter the value to subtract:
4
Enter your ID:
1
-2
The result is: -2

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
1
-2
The result is: -2

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```



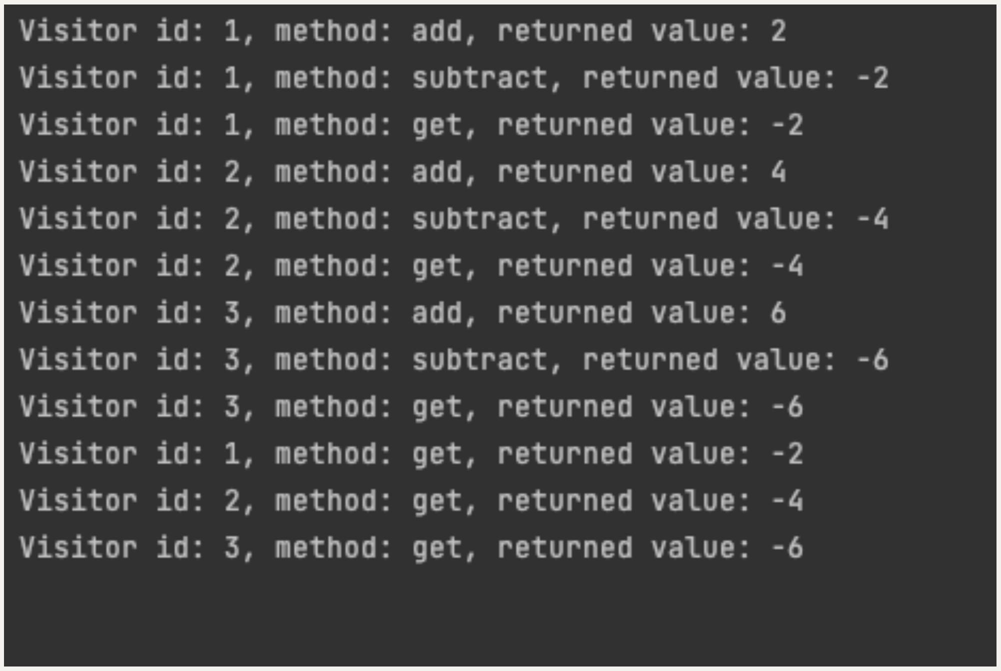
```
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
1
Enter the value to add:
6
Enter your ID:
3
6
The result is: 6

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
2
Enter the value to subtract:
12
Enter your ID:
3
-6
The result is: -6

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
3
-6
The result is: -6

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```



```
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
1
Enter the value to add:
4
Enter your ID:
2
4
The result is: 4

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
2
Enter the value to subtract:
8
Enter your ID:
2
-4
The result is: -4

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
2
-4
The result is: -4

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```


```
The client is running.
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
1
-2
The result is: -2

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```

```
The client is running.
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
3
-6
The result is: -6

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```

```
The client is running.
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:
2
-4
The result is: -4

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
4

Process finished with exit code 0
```

Project2Task4ServerConsole


```
Visitor id: 1, method: add, returned value: 2
Visitor id: 1, method: subtract, returned value: -2
Visitor id: 1, method: get, returned value: -2
Visitor id: 2, method: add, returned value: 4
Visitor id: 2, method: subtract, returned value: -4
Visitor id: 2, method: get, returned value: -4
Visitor id: 3, method: add, returned value: 6
Visitor id: 3, method: subtract, returned value: -6
Visitor id: 3, method: get, returned value: -6
Visitor id: 1, method: get, returned value: -2
Visitor id: 2, method: get, returned value: -4
Visitor id: 3, method: get, returned value: -6
```

Project 2 Task 5

Project2Task5Client

```
public class SigningClientTCP {
    public Socket socket;
    public int portNum;
```

```java
    RSA rsa;
    String id;
    Scanner scanner;
    public SigningClientTCP() {
        System.out.println("The client is running.");
        rsa = new RSA(); // create a rsa key pair
        System.out.println("Please enter server port: ");
        scanner = new Scanner(System.in);
        portNum = scanner.nextInt();
        scanner.nextLine();

        try {
            id = Utils.getID(rsa.getE().toString()+
rsa.getN().toString());
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
    }
    public void init() {//start the tcp client
        String s;
        while (true) {
            System.out.println("""
                    1. Add a value to your sum.
                    2. Subtract a value from your sum.
                    3. Get your sum.
                    4. Exit client""");
            s = scanner.nextLine();
            int choice = Integer.parseInt(s);

            int num;
            int res;

            if (choice == 1) {
                System.out.println("Enter the value to add: ");
                num = Integer.parseInt(scanner.nextLine());
                res = add(num, id);
            } else if (choice == 2) {
                System.out.println("Enter the value to subtract:");
                num = Integer.parseInt(scanner.nextLine());
                res = subtract(num, id);
            } else if (choice == 3) {
                System.out.println("Enter your ID: ");
                res = get(id);
            } else {
                return;
            }

            System.out.println("The result is: " + res);
            System.out.println();
        }
    }
    public static void main(String[] args) {
        SigningClientTCP client = new SigningClientTCP();
        client.init();
    }

    public int add(int i, String id) {
        String request = "add,"+i+","+id;
        return request(request);
    }

    public int subtract(int i, String id) {
        String request = "subtract,"+i+","+id;
        return request(request);
    }

    public int get(String id) {
        String request = "get,"+0+","+id;
        return request(request);
    }

    public int request(String requestBody) {
        BigInteger signature = rsa.sign(requestBody);
        String request =  requestBody
+";"+signature+";"+rsa.publicKey();
        try {
            socket = new Socket("localhost", portNum);
            BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter out = new PrintWriter(new
BufferedWriter(new
OutputStreamWriter(socket.getOutputStream())));
            out.println(request);
            out.println();
            out.flush();
            String res = in.readLine();
            System.out.println(res);
            return Integer.parseInt(res);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return 0;
    }
}


//source: https://github.com/CMU-Heinz-95702/Project-2-Client-
Server
public class RSA {
    private BigInteger n; // n is the modulus for both the
private and public keys
    private BigInteger e; // e is the exponent of the public key
    private BigInteger d; // d is the exponent of the private key
    private MessageDigest md;
    public BigInteger getN() {
        return n;
    }
```

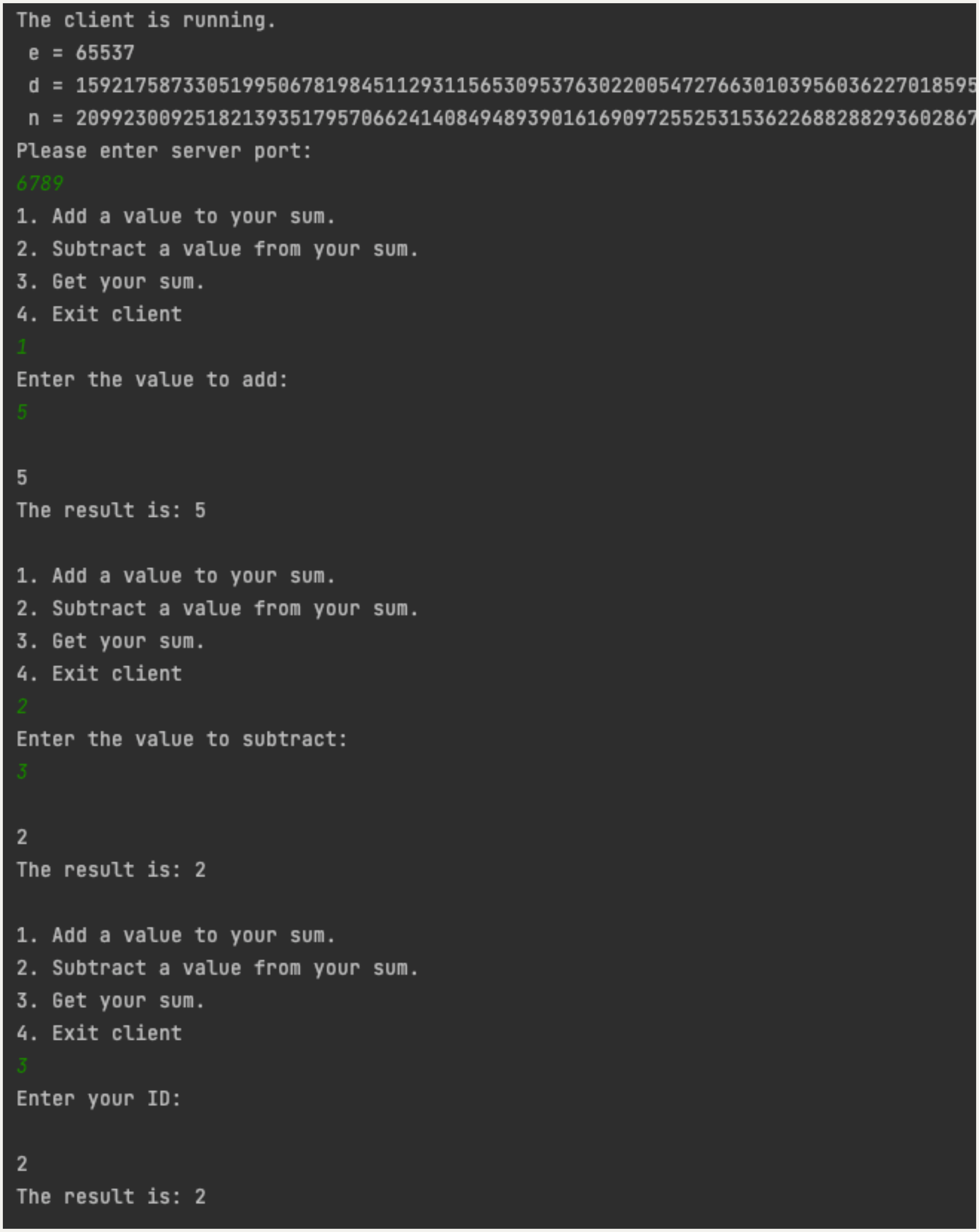
```java

    public BigInteger getE() {
        return e;
    }

    public BigInteger getD() {
        return d;
    }

    public String publicKey() {
        return e+","+n;
    }

    public BigInteger encryptWithPrivate(String message) {
        //use the private key to encrypt the message, add leading 0
        byte[] messageBytes = message.getBytes(StandardCharsets.UTF_8);
        byte[] signBytes = new byte[messageBytes.length+1];
        signBytes[0] = 0;
        System.arraycopy(messageBytes, 0, signBytes, 1, signBytes.length - 1);
        BigInteger m = new BigInteger(signBytes);
        return m.modPow(d, n);
    }

    public BigInteger sign(String message) {
        //encrypt the message with private key to sign
        byte[] m = Utils.getHashBytes(message);
        return encryptWithPrivate(Utils.bytesToHex(m));
    }

    public String decryptMessage(BigInteger message, BigInteger e, BigInteger n) {
        //decrypt the message with key pari e + n
        byte[] mb = message.modPow(e, n).toByteArray();
        return new String(mb);
    }

    public RSA() {
        Random rnd = new Random();

        // Generate two large random primes.
        BigInteger p = new BigInteger(400, 100, rnd);
        BigInteger q = new BigInteger(400, 100, rnd);

        // Compute n by the equation n = p * q.
        n = p.multiply(q);

        // Compute phi(n) = (p-1) * (q-1)
        BigInteger phi =
(p.subtract(BigInteger.ONE)).multiply(q.subtract(BigInteger.ONE))
;

        // Select a small odd integer e that is relatively prime
to phi(n).
        e = new BigInteger("65537");

        // Compute d as the multiplicative inverse of e modulo
phi(n).
        d = e.modInverse(phi);

        try { // for hash
            md = MessageDigest.getInstance("SHA-256");
        } catch (NoSuchAlgorithmException
noSuchAlgorithmException) {
            noSuchAlgorithmException.printStackTrace();
        }

        System.out.println(" e = " + e);
        System.out.println(" d = " + d);
        System.out.println(" n = " + n);
    }
}

//source: https://github.com/CMU-Heinz-95702/Project-2-Client-Server
//source: stack overflow
public class Utils {
    private static MessageDigest md;

    static {
        try {
            md = MessageDigest.getInstance("SHA-256");
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
    }

    private static final char[] HEX_ARRAY = "0123456789ABCDEF".toCharArray();
    public static String bytesToHex(byte[] bytes) {//convert byte to hex string
        char[] hexChars = new char[bytes.length * 2];
        for (int j = 0; j < bytes.length; j++) {
            int v = bytes[j] & 0xFF;
            hexChars[j * 2] = HEX_ARRAY[v >>> 4];
            hexChars[j * 2 + 1] = HEX_ARRAY[v & 0x0F];
        }
        return new String(hexChars);
    }

    public static byte[] getHashBytes(String message) {//get the hashed byte of a string
        return md.digest(message.getBytes(StandardCharsets.UTF_8));
    }

    public static String getID(String s) throws
NoSuchAlgorithmException {//get an ID using has
        byte[] hash = Utils.getHashBytes(s);
        byte[] idCode = new byte[20];
        System.arraycopy(hash, 0, idCode, 0, 20);
        return Utils.bytesToHex(idCode);
    }
}
```

Project2Task5Server

```
public class VerifyingServerTCP {
    public TreeMap<String, Integer> map;
    Socket socket;
    ServerSocket listenSocket;
    RSA rsa;

    public VerifyingServerTCP() {
        map = new TreeMap<>();
        socket = null;
        listenSocket = null;
        rsa = new RSA();

        try {
            listenSocket = new ServerSocket(6789);
        } catch (IOException e) {
            e.printStackTrace();
        }

    }

    public void init() {//start the server
        while (true) {
            try {
                socket = listenSocket.accept();
```

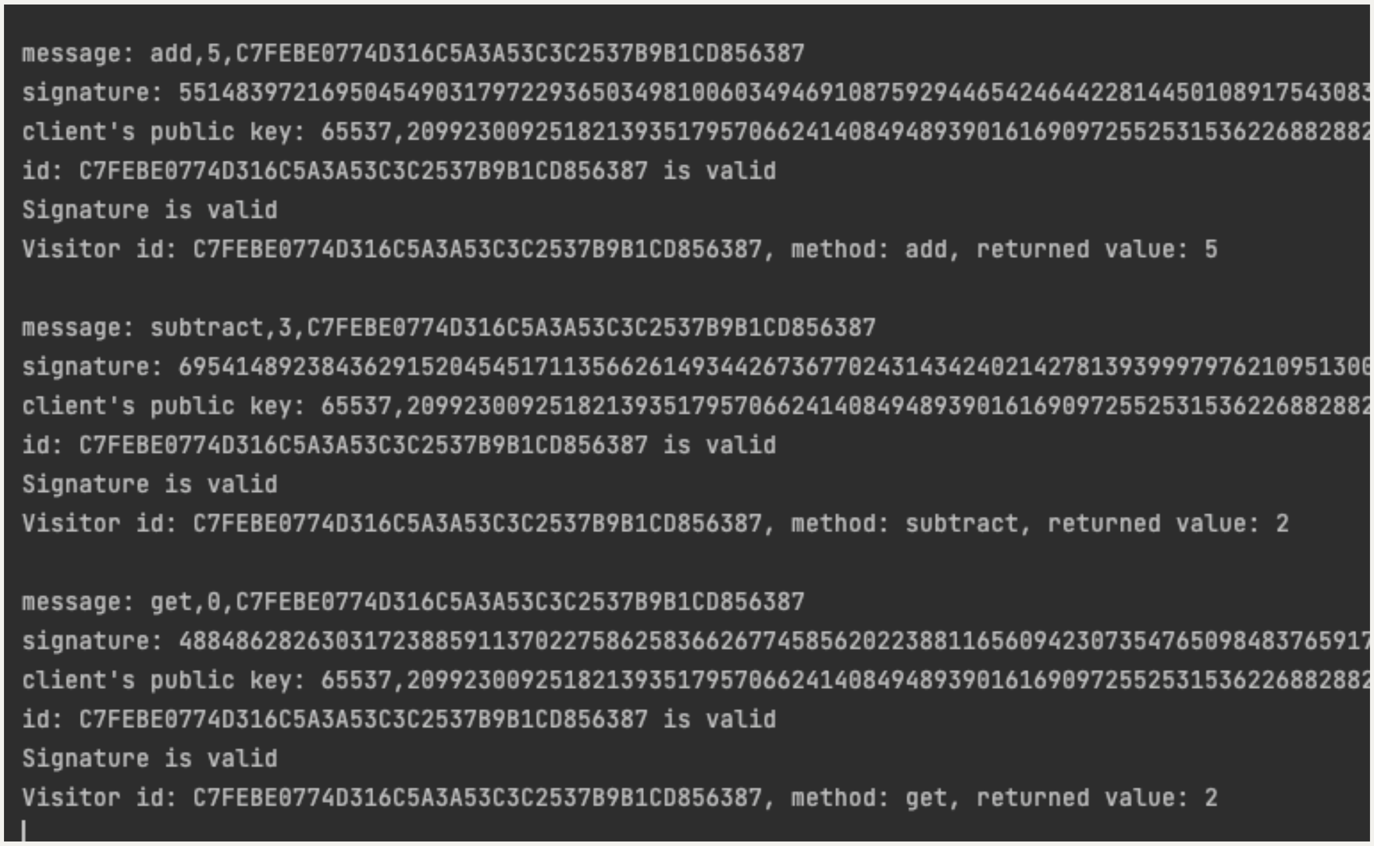
```java
                Scanner in = new Scanner(socket.getInputStream());
                StringBuilder request = new StringBuilder();
                String s;
                while (in.hasNextLine()) {
                    s = in.nextLine();
                    if (s.equals("")) {
                        PrintWriter out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(socket.getOutputStream())));
                        String requestBody = request.toString();
                        requestBody = verify(requestBody); // verify the message get the real request
                        if (requestBody.equals("Error in request")) {
                            out.println("Error in request");
                            break;
                        }
                        int sum = exec(requestBody);

                        byte[] res = String.valueOf(sum).getBytes();
                        out.println(new String(res));
                        out.flush();
                        socket.close();
                    } else {
                        request.append(s);
                    }
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }

    public String verify(String request) {//verify the request and return the real request info
        //display all the info of client
        String[] s = request.split(";");
        String message = s[0];
        String signature = s[1];
        String[] pubKey = s[2].split(",");
        BigInteger e = new BigInteger(pubKey[0]);
        BigInteger n = new BigInteger(pubKey[1]);
        System.out.println();
        System.out.println("message: " + message);
        System.out.println("signature: " + signature);
        System.out.println("client's public key: " + e+","+n);

        String id = message.split(",")[2];
        String testId = "";
        try {//use the public key to verify the id
            testId = Utils.getID(e.toString()+n.toString());
        } catch (NoSuchAlgorithmException noSuchAlgorithmException) {
            noSuchAlgorithmException.printStackTrace();
        }

        //decrypt the message to get the hashed string
        String testMessage = rsa.decryptMessage(new BigInteger(signature), e, n);

        //hashed the message
        String hashMessage = Utils.bytesToHex(Utils.getHashBytes(message));

        if (id.equals(testId)) {
            System.out.println("id: "+ testId +" is valid");
        } else {
            return "Error in request";
        }
        if (testMessage.equals(hashMessage)) {
            System.out.println("Signature is valid");
        } else {
            return "Error in request";
        }
        return message;
    }

    public int exec(String requestBody) {
        String[] params = requestBody.split(",");
        String method = params[0];
        int num = Integer.parseInt(params[1]);
        String id = params[2];
        if (!map.containsKey(id)) map.put(id, 0);
        int res;
        if (method.equals("get")) {
            res = get(id);
        } else if (method.equals("add")) {
            res = add(id, num);
        } else {
            res = subtract(id, num);
        }
        System.out.printf("Visitor id: %s, method: %s, returned value: %d\n", id, method, res);
        return map.get(id);
    }

    public int add(String id, int num) {
        map.put(id,map.get(id)+num);
        return map.get(id);
    }

    public int subtract(String id, int num) {
        map.put(id,map.get(id)-num);
        return map.get(id);
    }

    public int get(String id) {
        return map.get(id);
    }

    public static void main(String[] args) {
        VerifyingServerTCP server = new VerifyingServerTCP();
        server.init();
    }
}

```

Project2Task5ClientConsole



```
The client is running.
 e = 65537
 d = 15921758733051995067819845112931156530953763022005472766301039560362270185 95
 n = 20992300925182139351795706624140849489390161690972552531536226882882936028 67
Please enter server port:
6789
1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
1
Enter the value to add:
5


5
The result is: 5

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
2
Enter the value to subtract:
3


2
The result is: 2

1. Add a value to your sum.
2. Subtract a value from your sum.
3. Get your sum.
4. Exit client
3
Enter your ID:

2
The result is: 2
```

Project2Task5ServerConsole



```
message: add,5,C7FEBE0774D316C5A3A53C3C2537B9B1CD856387
signature: 5514839721695045490317972293650349810060349469108759294465424644228144501089175430830
client's public key: 65537,2099230092518213935179570662414084948939016169097255253153622688288282
id: C7FEBE0774D316C5A3A53C3C2537B9B1CD856387 is valid
Signature is valid
Visitor id: C7FEBE0774D316C5A3A53C3C2537B9B1CD856387, method: add, returned value: 5

message: subtract,3,C7FEBE0774D316C5A3A53C3C2537B9B1CD856387
signature: 69541489238436291520454517113566261493442673677024314342402142781393999797621095130
client's public key: 65537,2099230092518213935179570662414084948939016169097255253153622688288282
id: C7FEBE0774D316C5A3A53C3C2537B9B1CD856387 is valid
Signature is valid
Visitor id: C7FEBE0774D316C5A3A53C3C2537B9B1CD856387, method: subtract, returned value: 2

message: get,0,C7FEBE0774D316C5A3A53C3C2537B9B1CD856387
signature: 48848628263031723885911370227586258366267745856202238811656094230735476509848376591
client's public key: 65537,2099230092518213935179570662414084948939016169097255253153622688288282
id: C7FEBE0774D316C5A3A53C3C2537B9B1CD856387 is valid
Signature is valid
Visitor id: C7FEBE0774D316C5A3A53C3C2537B9B1CD856387, method: get, returned value: 2
```