Computer Networks Assignment - 11 Socket Programming

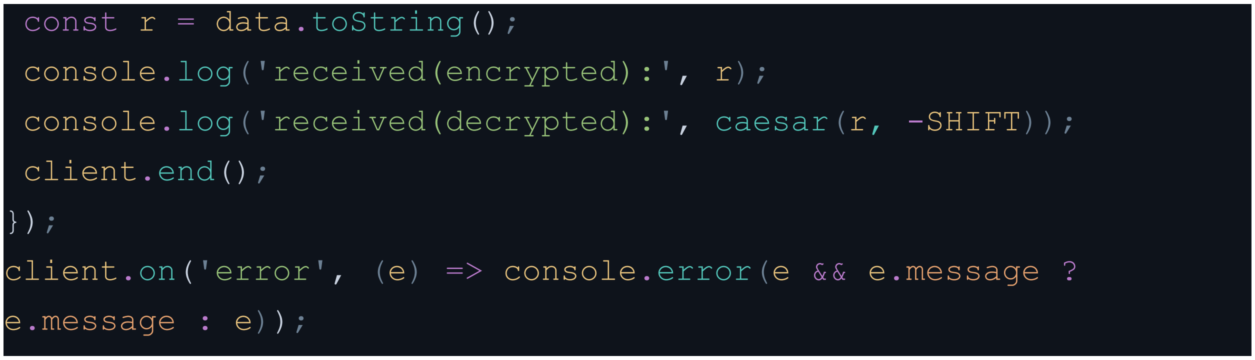
Praneesh R V
CB.SC.U4CYS23036

Server.js

```
const net = require('net');
const port = 1400;
const host = '127.0.0.1';
const SHIFT = Number.isFinite(parseInt(process.argv[2], 10)) ?
parseInt(process.argv[2], 10) : 13;
function caesar(text, shift) {
 const s = ((shift % 26) + 26) % 26;
 return text
   .split('')
   .map((ch) => {
     const c = ch.charCodeAt(0);
     if (c >= 97 && c <= 122) return String.fromCharCode(((c -
97 + s) % 26) + 97);
     if (c >= 65 && c <= 90) return String.fromCharCode(((c - 65
+ s) % 26) + 65);
     return ch;
   })
   .join('');
}
const server = net.createServer((socket) => {
 console.log('client connected');
 socket.on('data', (buf) => {
   const msg = buf.toString();
   console.log('received(encrypted):', msg);
   console.log('received(decrypted):', caesar(msg, -SHIFT));
   socket.write(caesar('connected', SHIFT));
 });
});
server.listen(port, host, () => console.log('server listening
on', host, ':', port));
```

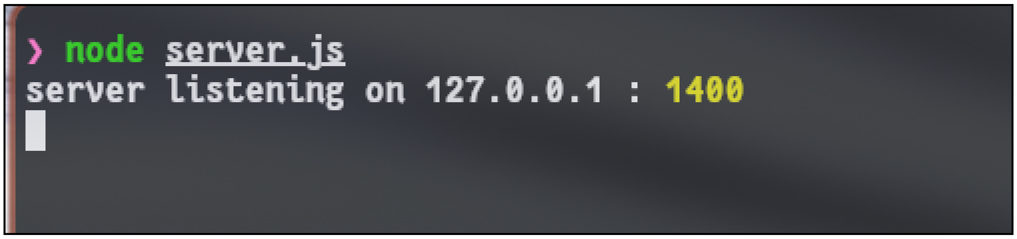
Client.js

```
const net = require('net');
const client = new net.Socket();
const port = 1400;
const host = '127.0.0.1';
const SHIFT = Number.isFinite(parseInt(process.argv[2], 10)) ?
parseInt(process.argv[2], 10) : 13;
function caesar(text, shift) {
 const s = ((shift % 26) + 26) % 26;
 return text
   .split('')
   .map((ch) => {
     const c = ch.charCodeAt(0);
     if (c >= 97 && c <= 122) return String.fromCharCode(((c -
97 + s) % 26) + 97);
     if (c >= 65 && c <= 90) return String.fromCharCode(((c - 65
+ s) % 26) + 65);
     return ch;
   })
   .join('');
}
client.connect(port, host, () => {
 const p = 'Hello';
 const enc = caesar(p, SHIFT);
 console.log('connected -> sending(encrypted):', enc);
 client.write(enc);
});
client.on('data', (data) => {
```



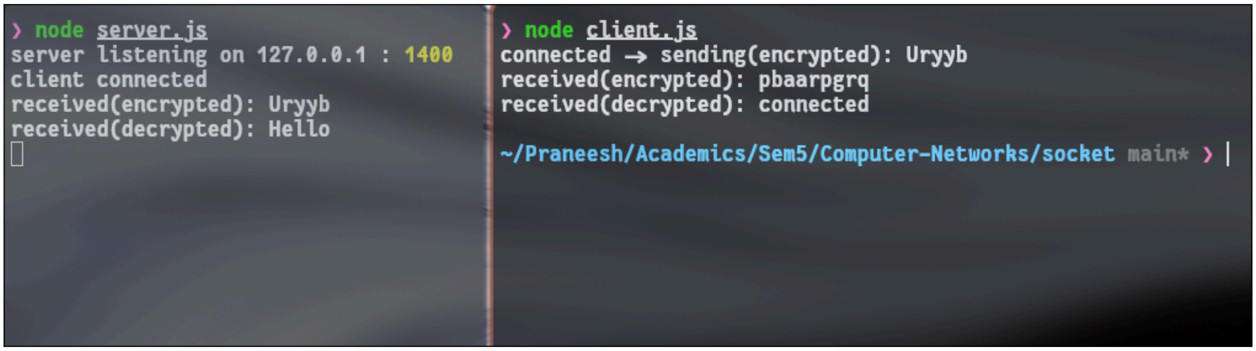
```
 const r = data.toString();
 console.log('received(encrypted):', r);
 console.log('received(decrypted):', caesar(r, -SHIFT));
 client.end();
});
client.on('error', (e) => console.error(e && e.message ?
e.message : e));
```

Starting the server



```
❯ node server.js
server listening on 127.0.0.1 : 1400
```

Once the client.js is invoked, it sends an encrypted message to the server, and the server too, sends an encrypted message to the client
It is a shift cipher with the shift being 13



```
❯ node server.js
server listening on 127.0.0.1 : 1400
client connected
received(encrypted): Uryyb
received(decrypted): Hello
```

```
❯ node client.js
connected → sending(encrypted): Uryyb
received(encrypted): pbaarpgrq
received(decrypted): connected

~/Praneesh/Academics/Sem5/Computer-Networks/socket main* ❯ |
```