Name: Preyash Date: 27-01-2022

Registration Number: 20BPS1022

**LAB-04**

**Socket Programming**

**AIM:** To perform the given tasks using the socket programming.

**Task 1:** Simulate an online English dictionary using sockets. A user searches for meaning of a word in the dictionary which is present in the server. The server responds to the user with the corresponding meaning of the word.

**ALGORITHM**

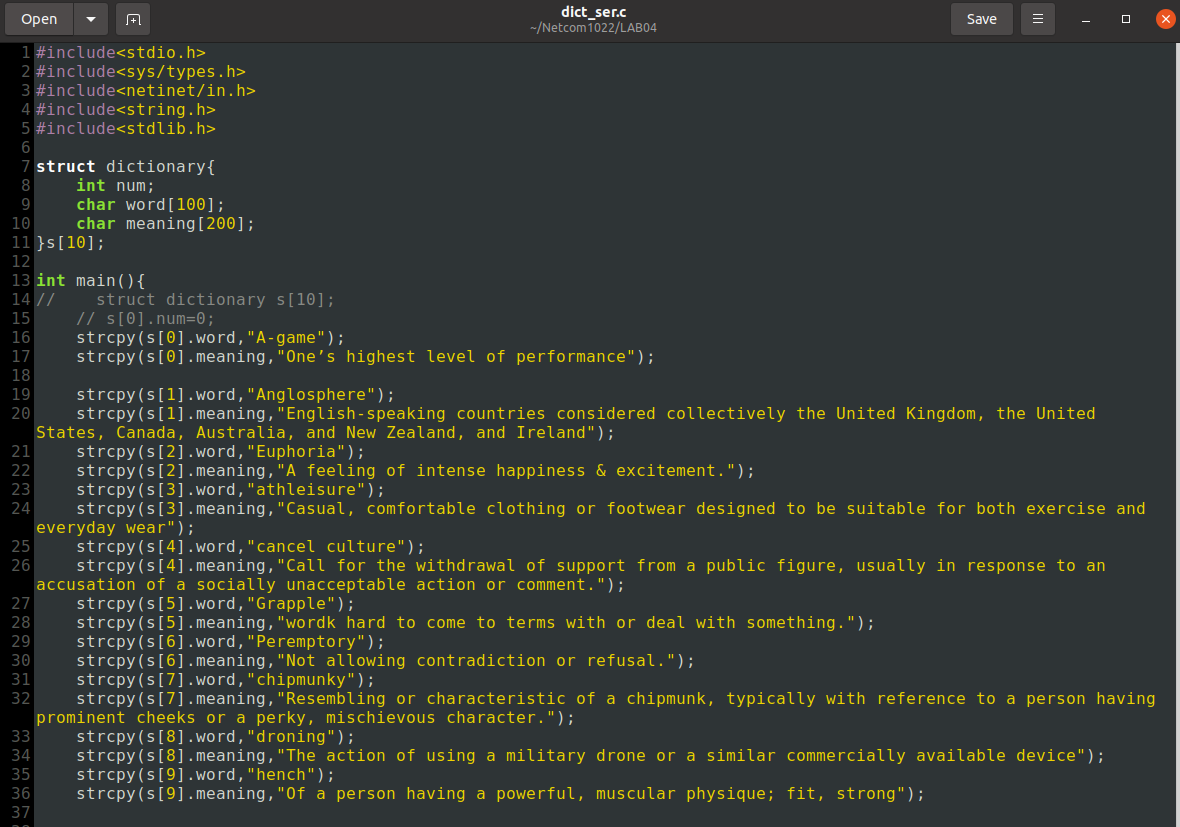
**Server-side Algorithm**

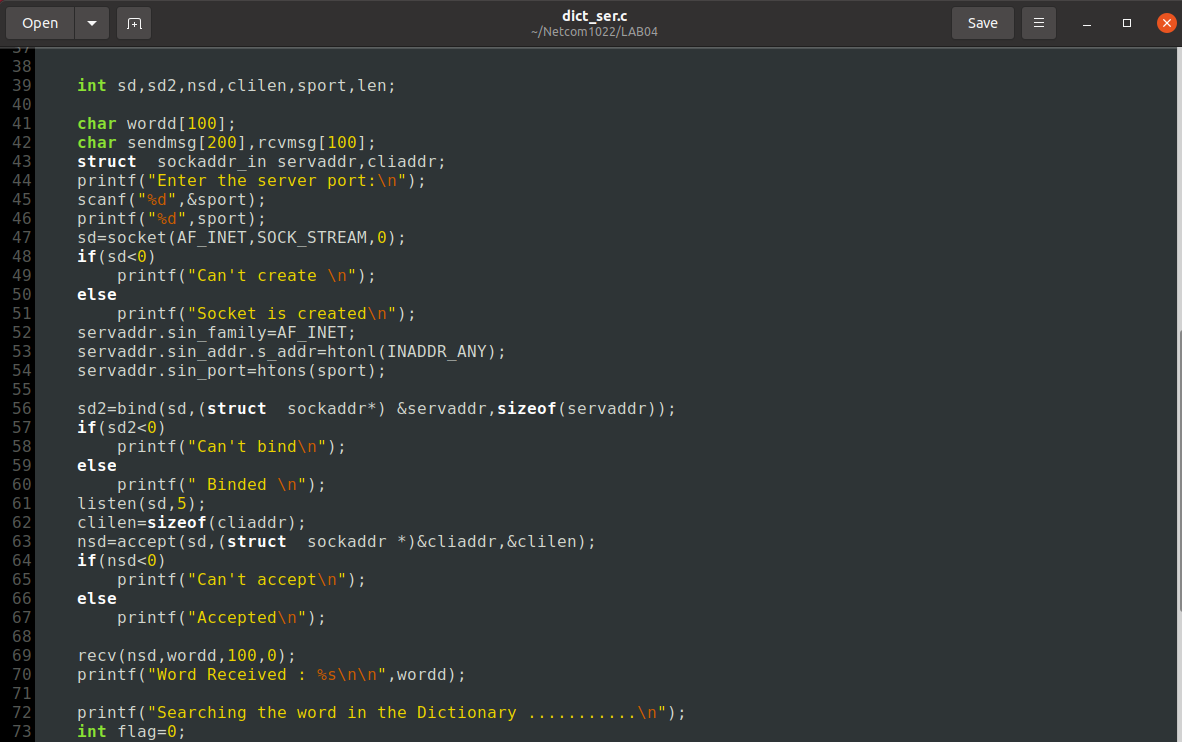
1. Begin by declaring variables.
2. Take the user's input for the port number.
3. Create a TCP socket for the server with socket().
4. Bind the socket to the server address with bind().
5. listen() places the server socket in passive mode, waiting for a client to reach it.
6. accept() establishes a connection between the client and the server in order to send data.
7. Sending a message to the client indicating the connection has been established.
8. Make a structure declaration. The word is stored as a word variable in dictionary, and the variable is stored in mean. Create a total of ten objects and define their variables.
9. Receive the word from the client and see whether it's in any of the objects; if it is, copy the meaning and send it to the msgvariable; if it's not, copy "Meaning not found."
10. Send a message to the client.
11. Stop the programme.

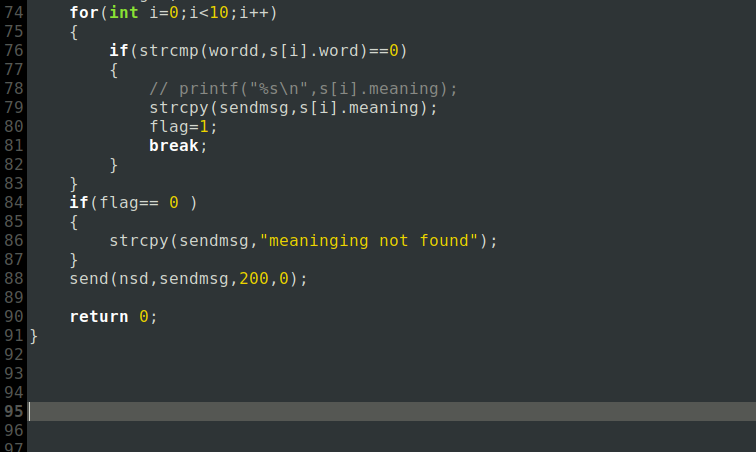
**Client-Side Algorithm**

1. Begin by declaring variables.
2. Take the user's port as input.
3. Create a TCP socket for the client with socket().
4. Connect the client to the server using connect() to exchange data.
5. Recv() is used to receive data from the server.
6. The data will be restored if the connection is successful.
7. Enter the term you're looking for and submit it to the server.
8. Obtain the message from the server.
9. On the client side, print the definition of the term.
10. Put an end to the programme.

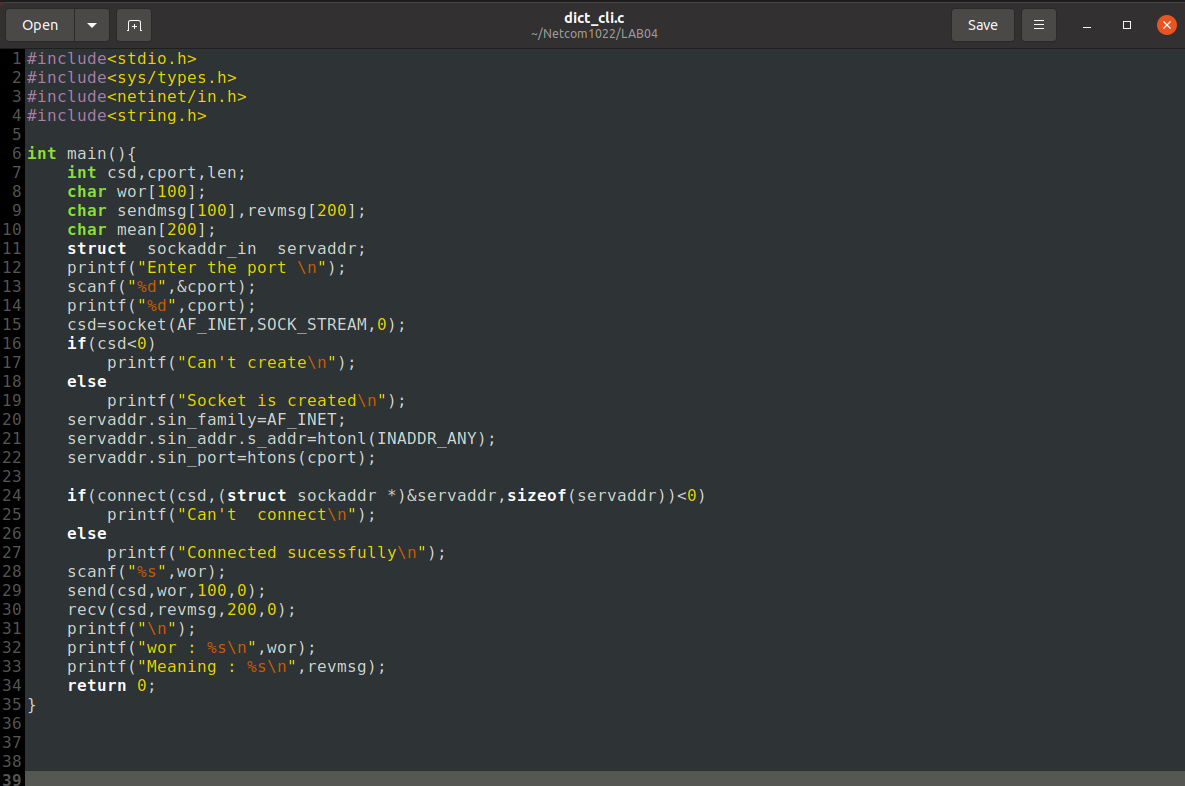
**Server Program Source Code:**

****

****

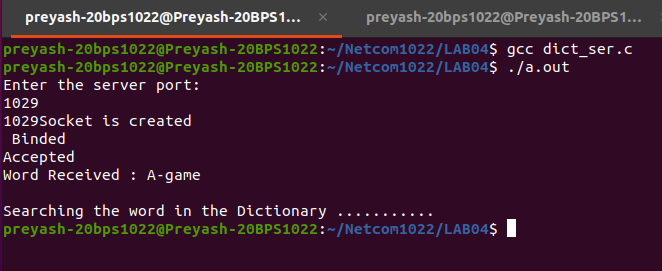
****

**Client Program Source Code:**

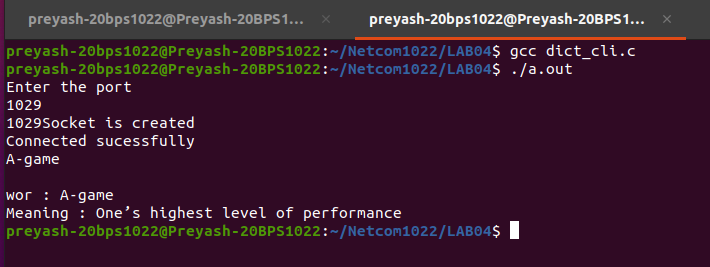
****

**Output:**

**Server-side:**

****

**Client-side:**

****

**Task 2:** Develop a “Remote Calculator” application that works as follows: The client program inputs two integers and an arithmetic operation (‘\*’;’/;’+’; ’-’) from the user and sends these three values to the server side. The server does the given operation or the two integers and sends back the result of the operation on the two integers and sends back the result of the operation to the client. Help server to implement this scenario using connection-oriented sockets.

**ALGORITHM**

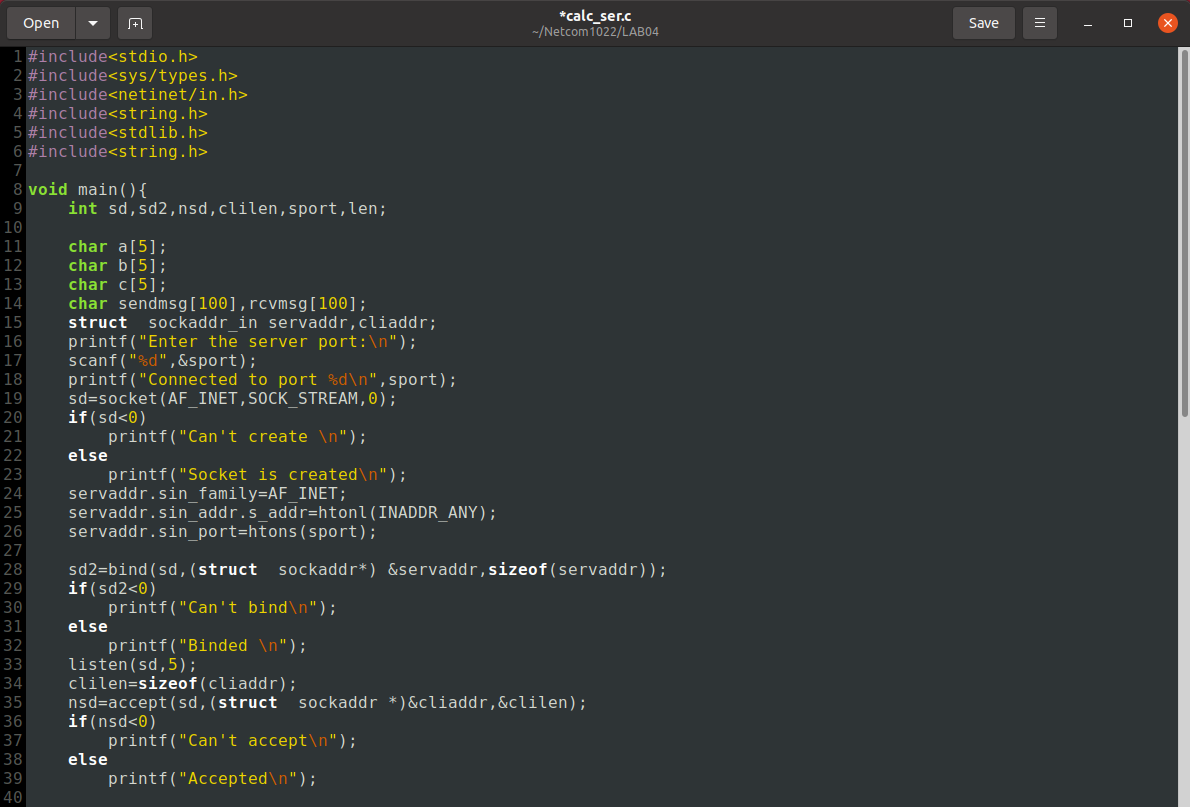
**Server-side Algorithm**

1. Begin by declaring variables.
2. Take the user's input for the port number.
3. Create a TCP socket for the server with socket().
4. Bind the socket to the server address with bind().
5. listen() places the server socket in passive mode, waiting for a client to reach it.
6. accept() establishes a connection between the client and the server in order to send data.
7. Sending a message to the client indicating the connection has been established.
8. From the server, get the two numbers and the operation you wish to execute.
9. Calculate the result as needed.
10. Send a message to the client.
11. Stop the programme.

**Client-Side Algorithm**

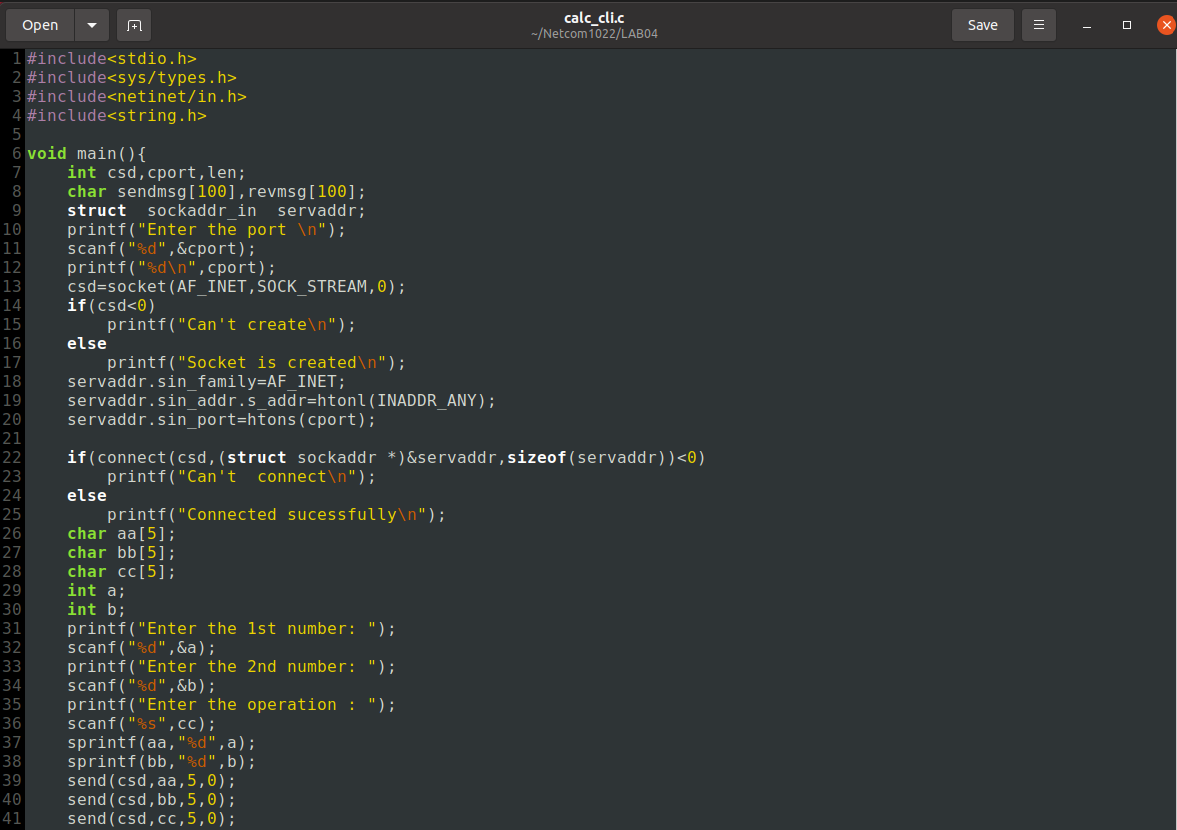
1. Begin by declaring variables.
2. Take the user's port as input.
3. Create a TCP socket for the client with socket().
4. Connect the client to the server using connect() to exchange data.
5. Recv() is used to receive data from the server.
6. Send the values to the server for calculation.
7. Receive the result from the server.
8. Print the output.
9. End the program.

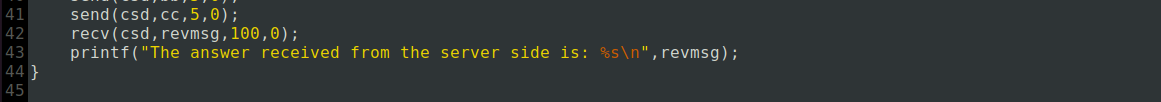
**Server Program Source Code:**

****

****

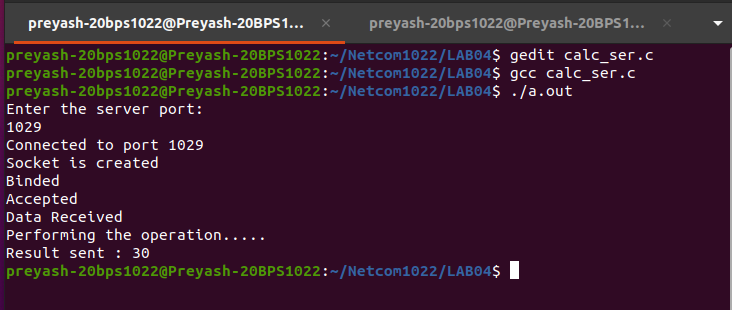
**Client Program Source Code:**

****

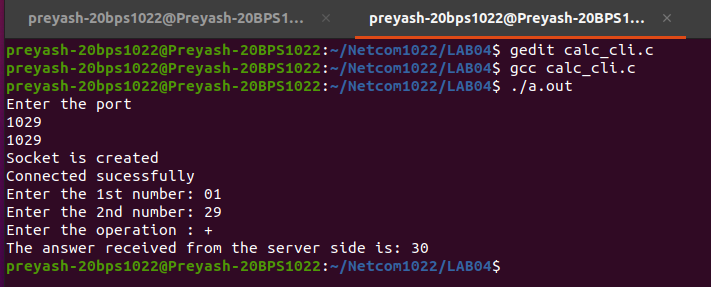
****

**Output:**

**Server-side:**

****

**Client-side:**

****