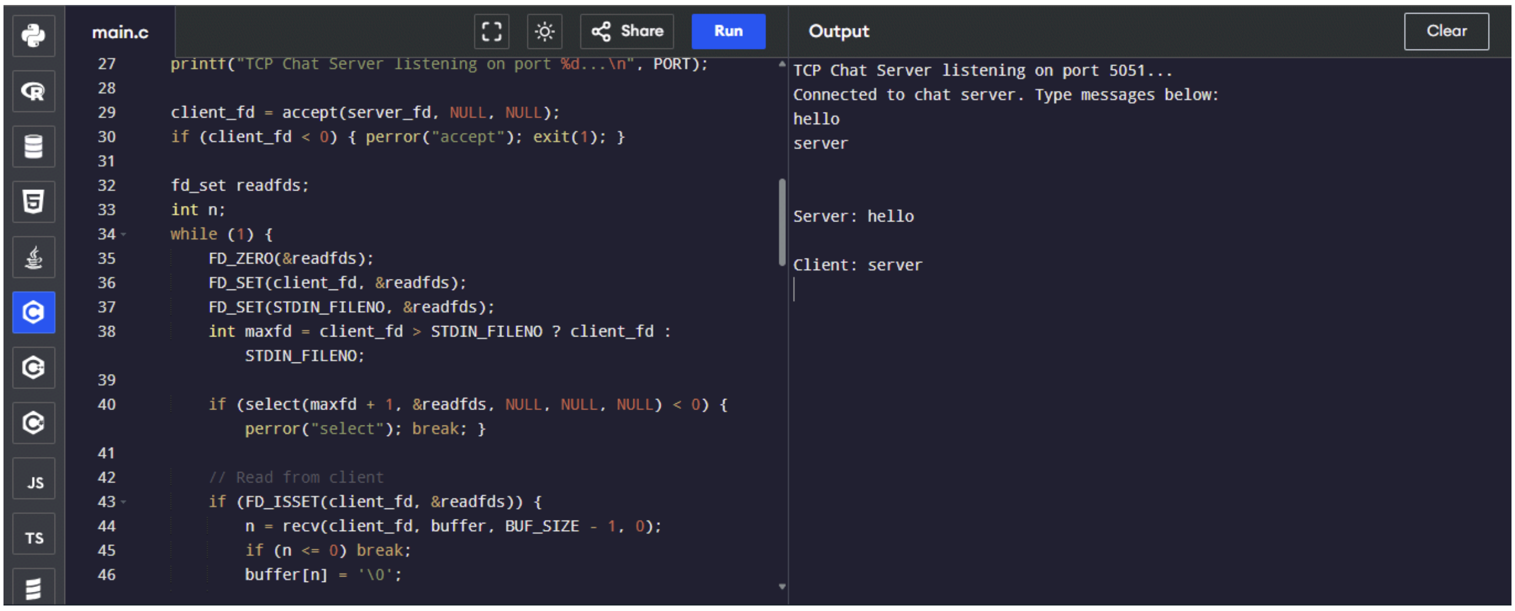
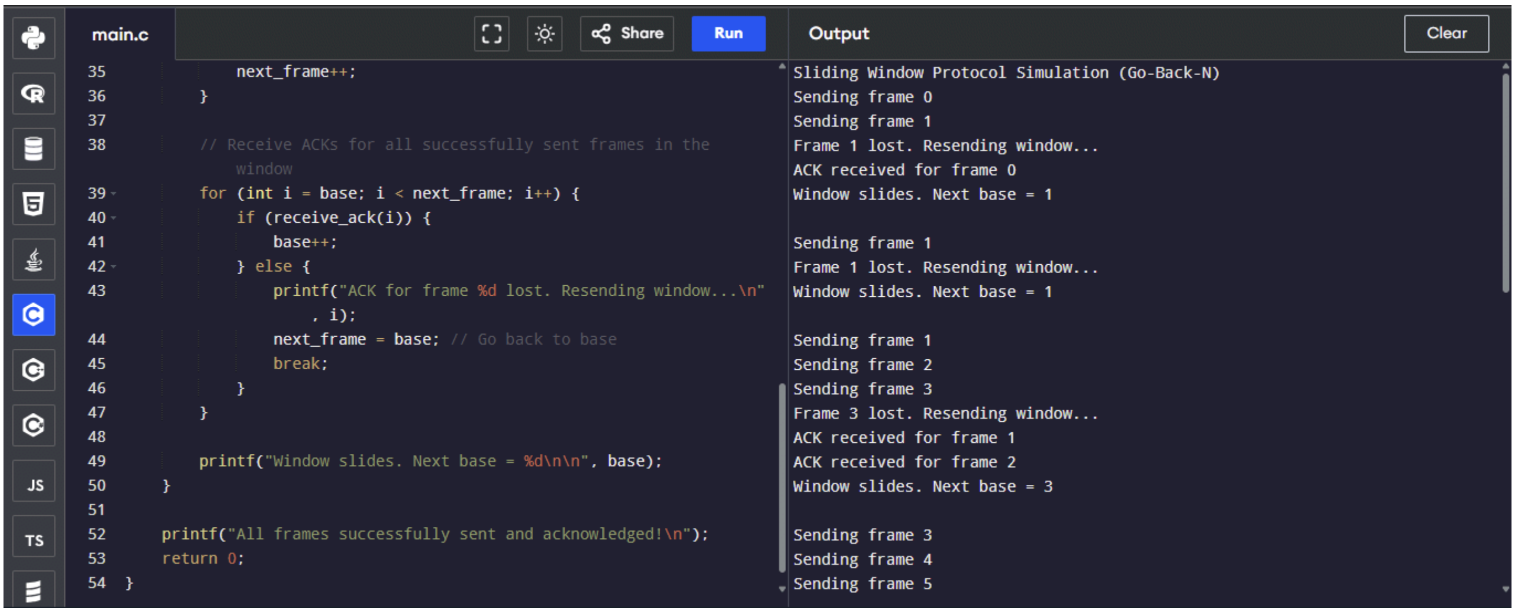


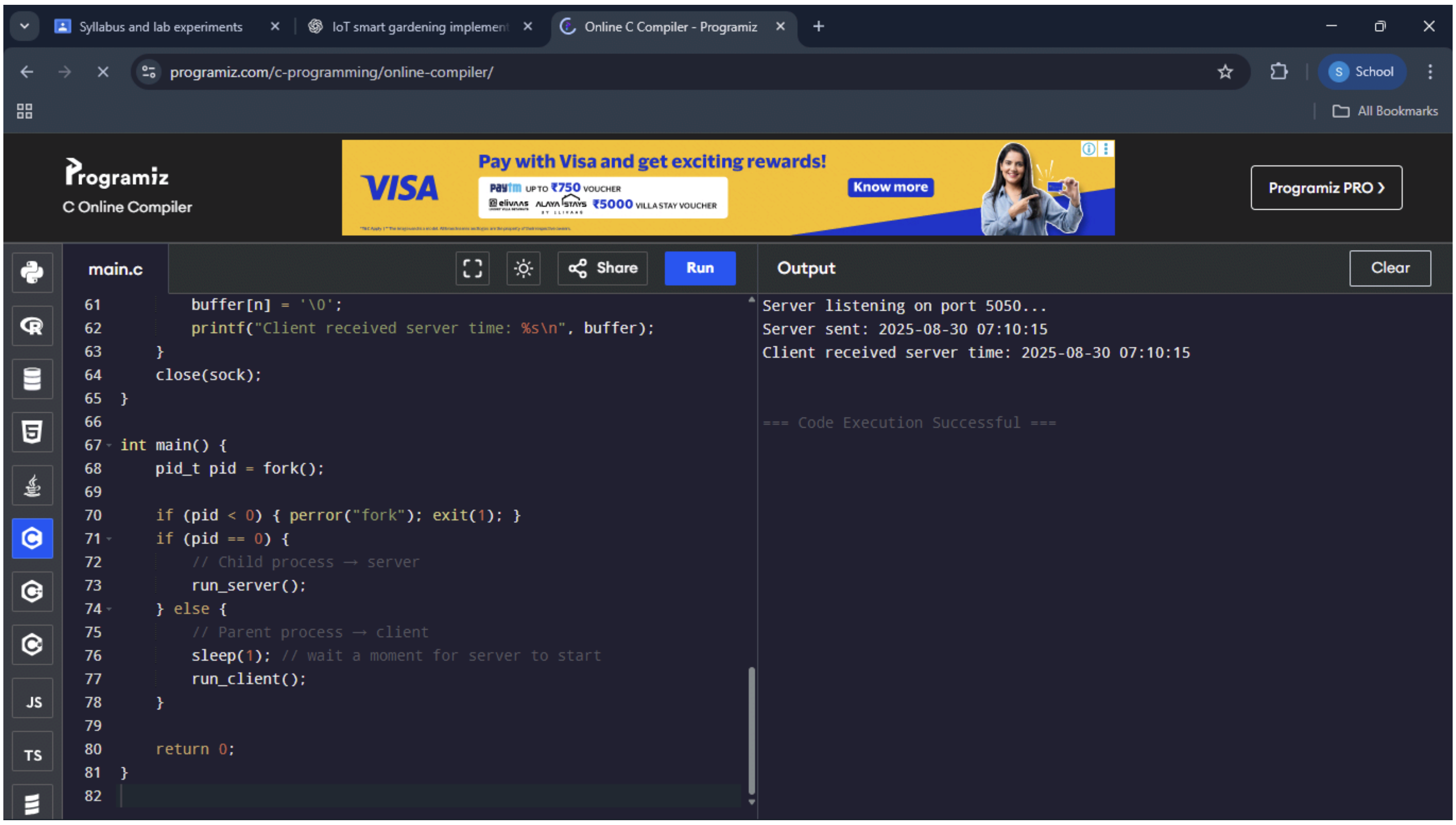
```c
printf("TCP Chat Server listening on port %d...\n", PORT);

client_fd = accept(server_fd, NULL, NULL);
if (client_fd < 0) { perror("accept"); exit(1); }

fd_set readfds;
int n;
while (1) {
    FD_ZERO(&readfds);
    FD_SET(client_fd, &readfds);
    FD_SET(STDIN_FILENO, &readfds);
    int maxfd = client_fd > STDIN_FILENO ? client_fd :
        STDIN_FILENO;

    if (select(maxfd + 1, &readfds, NULL, NULL, NULL) < 0) {
        perror("select"); break; }

    // Read from client
    if (FD_ISSET(client_fd, &readfds)) {
        n = recv(client_fd, buffer, BUF_SIZE - 1, 0);
        if (n <= 0) break;
        buffer[n] = '\0';
```

```
TCP Chat Server listening on port 5051...
Connected to chat server. Type messages below:
hello
server


Server: hello

Client: server
```



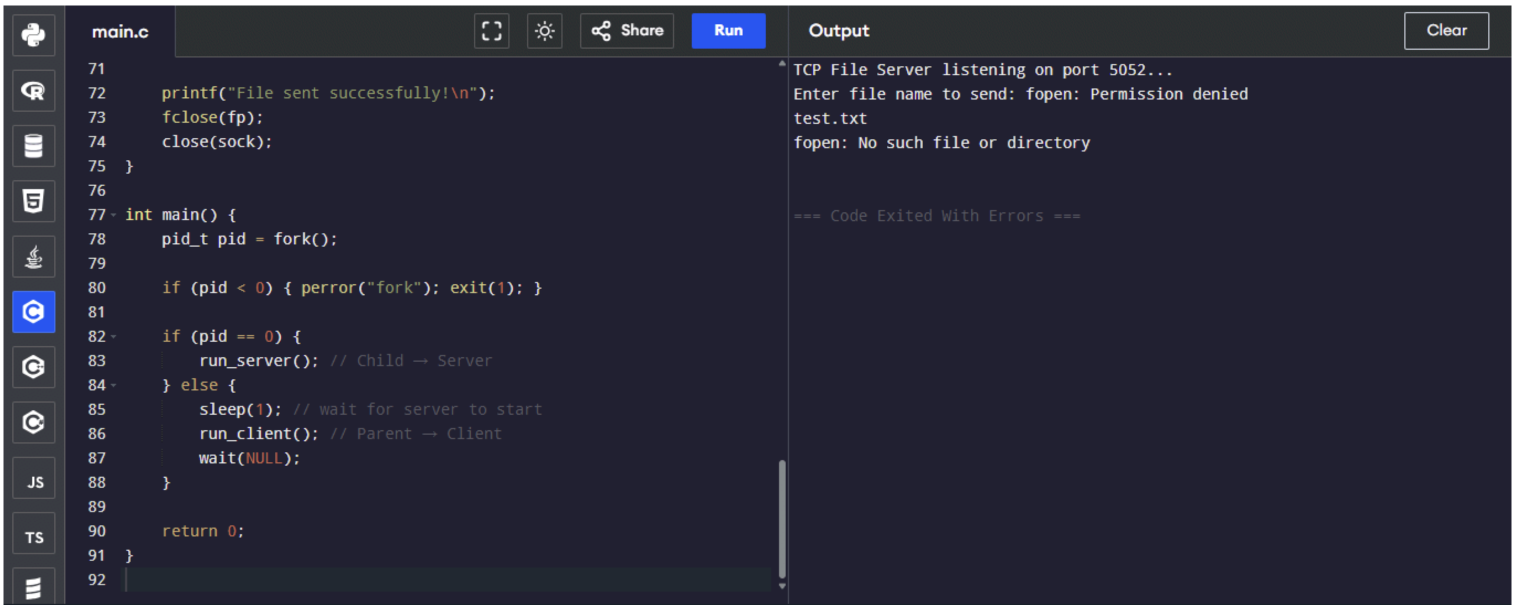
```c
            next_frame++;
        }

        // Receive ACKs for all successfully sent frames in the
            window
        for (int i = base; i < next_frame; i++) {
            if (receive_ack(i)) {
                base++;
            } else {
                printf("ACK for frame %d lost. Resending window...\n"
                    , i);
                next_frame = base; // Go back to base
                break;
            }
        }

        printf("Window slides. Next base = %d\n\n", base);
    }

    printf("All frames successfully sent and acknowledged!\n");
    return 0;
}
```

```
Sliding Window Protocol Simulation (Go-Back-N)
Sending frame 0
Sending frame 1
Frame 1 lost. Resending window...
ACK received for frame 0
Window slides. Next base = 1

Sending frame 1
Frame 1 lost. Resending window...
Window slides. Next base = 1

Sending frame 1
Sending frame 2
Sending frame 3
Frame 3 lost. Resending window...
ACK received for frame 1
ACK received for frame 2
Window slides. Next base = 3

Sending frame 3
Sending frame 4
Sending frame 5
```


Syllabus and lab experiments
IoT smart gardening implement
Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
School
All Bookmarks
Programiz
C Online Compiler
Pay with Visa and get exciting rewards!
Know more
Programiz PRO
main.c
Share
Run
Output
Clear

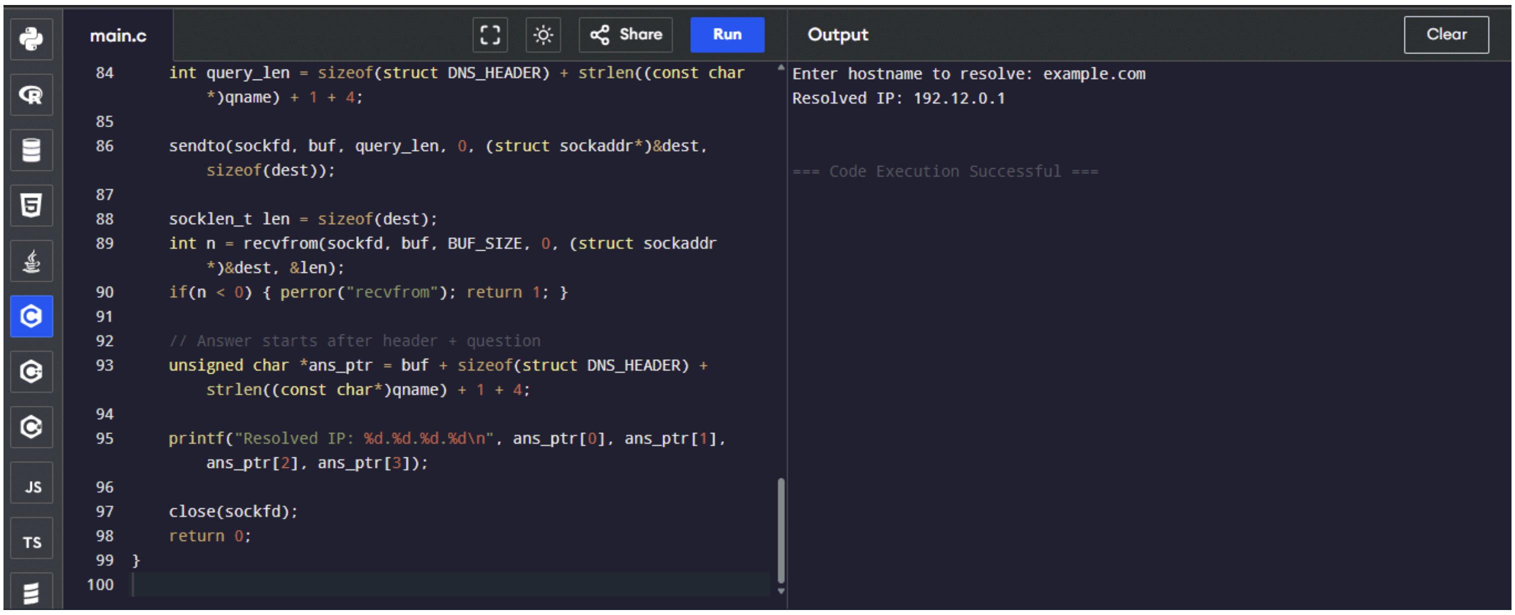
```c
        buffer[n] = '\0';
        printf("Client received server time: %s\n", buffer);
    }
    close(sock);
}

int main() {
    pid_t pid = fork();

    if (pid < 0) { perror("fork"); exit(1); }
    if (pid == 0) {
        // Child process → server
        run_server();
    } else {
        // Parent process → client
        sleep(1); // wait a moment for server to start
        run_client();
    }

    return 0;
}
```

```
Server listening on port 5050...
Server sent: 2025-08-30 07:10:15
Client received server time: 2025-08-30 07:10:15

=== Code Execution Successful ===
```



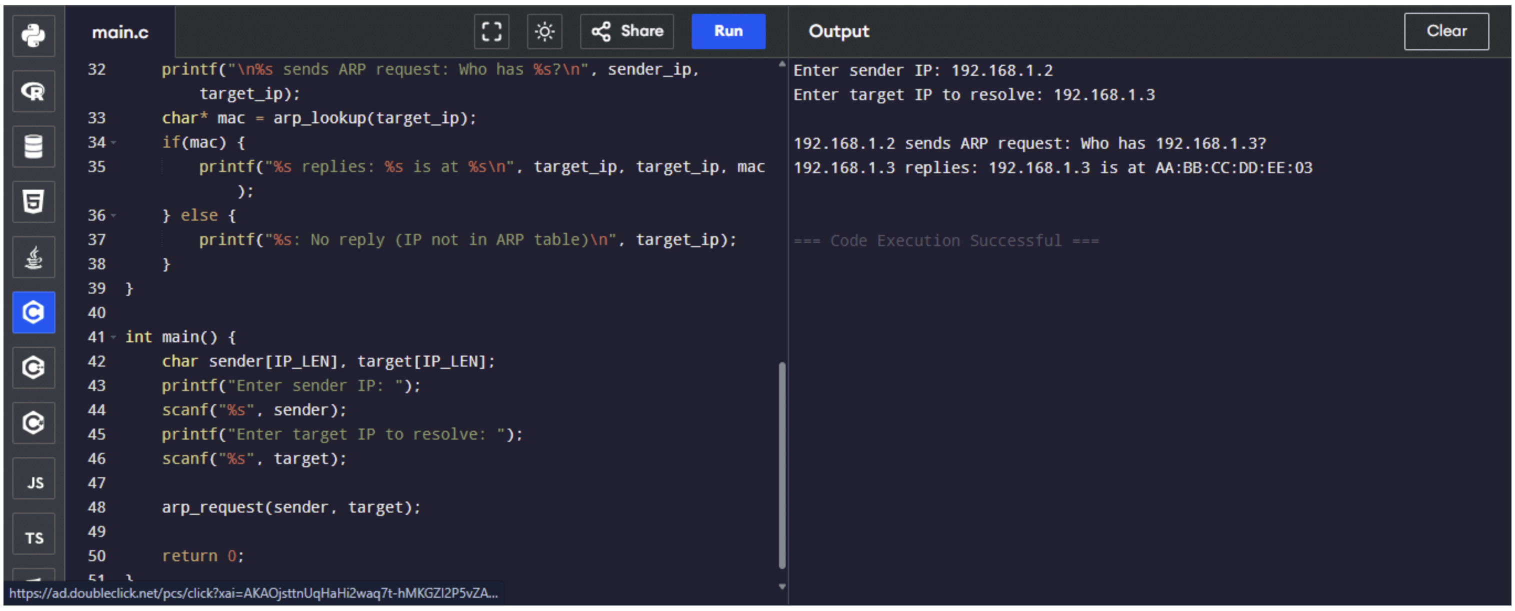
```c
    printf("File sent successfully!\n");
    fclose(fp);
    close(sock);
}

int main() {
    pid_t pid = fork();

    if (pid < 0) { perror("fork"); exit(1); }

    if (pid == 0) {
        run_server(); // Child → Server
    } else {
        sleep(1); // wait for server to start
        run_client(); // Parent → Client
        wait(NULL);
    }

    return 0;
}
```

```
TCP File Server listening on port 5052...
Enter file name to send: fopen: Permission denied
test.txt
fopen: No such file or directory


=== Code Exited With Errors ===
```



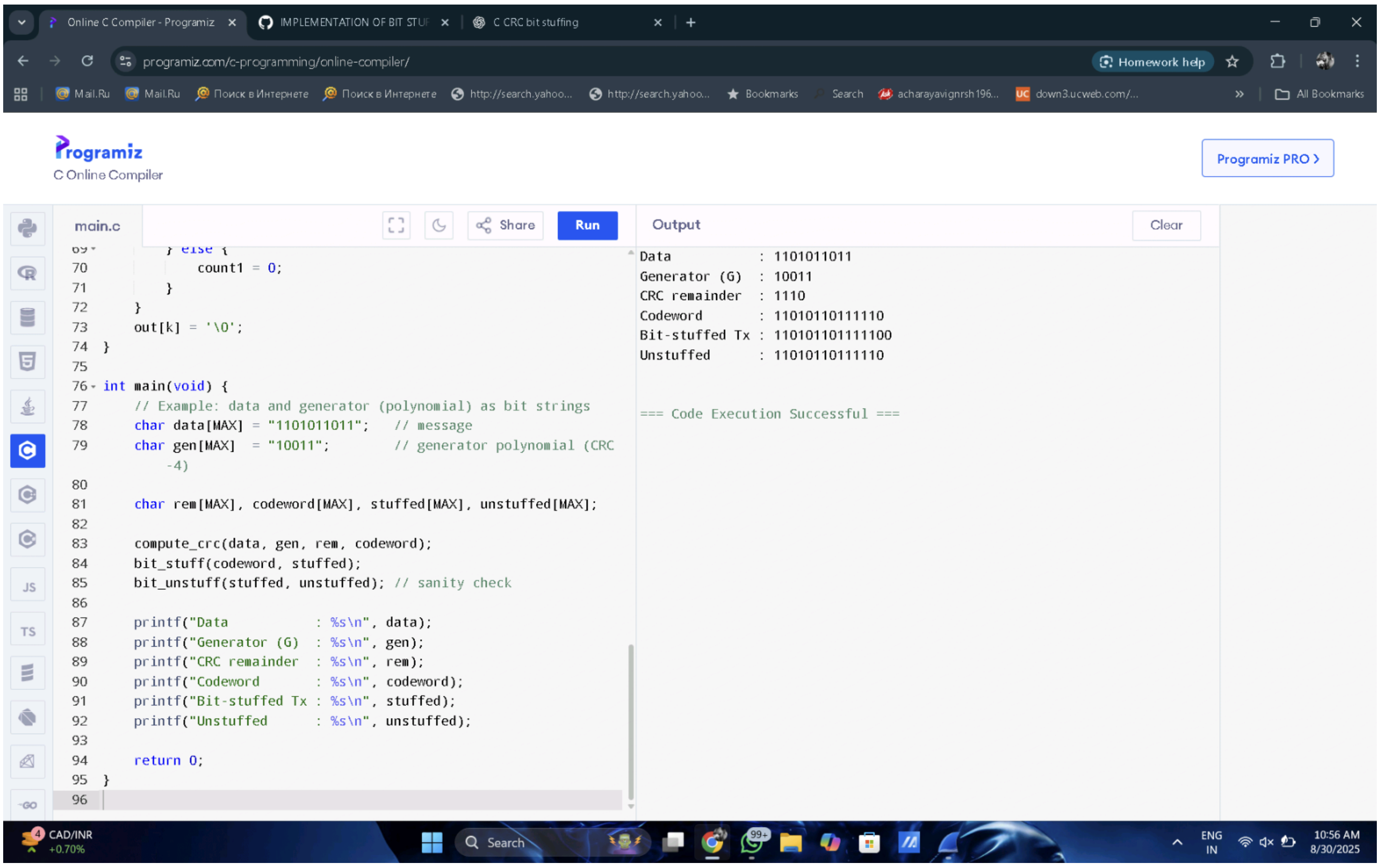
```c
    int query_len = sizeof(struct DNS_HEADER) + strlen((const char
        *)qname) + 1 + 4;

    sendto(sockfd, buf, query_len, 0, (struct sockaddr*)&dest,
        sizeof(dest));

    socklen_t len = sizeof(dest);
    int n = recvfrom(sockfd, buf, BUF_SIZE, 0, (struct sockaddr
        *)&dest, &len);
    if(n < 0) { perror("recvfrom"); return 1; }

    // Answer starts after header + question
    unsigned char *ans_ptr = buf + sizeof(struct DNS_HEADER) +
        strlen((const char*)qname) + 1 + 4;

    printf("Resolved IP: %d.%d.%d.%d\n", ans_ptr[0], ans_ptr[1],
        ans_ptr[2], ans_ptr[3]);

    close(sockfd);
    return 0;
}
```

```
Enter hostname to resolve: example.com
Resolved IP: 192.12.0.1


=== Code Execution Successful ===
```


main.c
Share
Run
Output
Clear
JS
TS
https://ad.doubleclick.net/pcs/click?xai=AKAOjsttnUqHaHi2waq7t-hMKGZI2P5vZA...

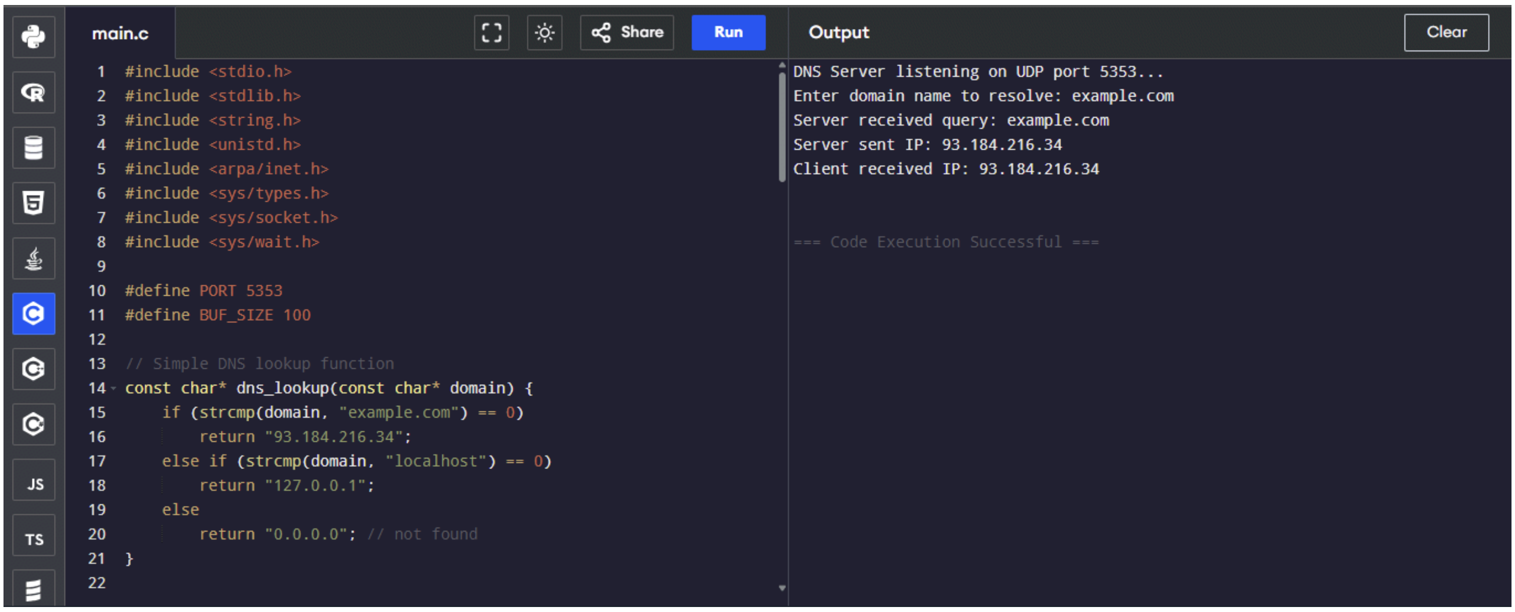
```c
    printf("\n%s sends ARP request: Who has %s?\n", sender_ip,
        target_ip);
    char* mac = arp_lookup(target_ip);
    if(mac) {
        printf("%s replies: %s is at %s\n", target_ip, target_ip, mac
            );
    } else {
        printf("%s: No reply (IP not in ARP table)\n", target_ip);
    }
}

int main() {
    char sender[IP_LEN], target[IP_LEN];
    printf("Enter sender IP: ");
    scanf("%s", sender);
    printf("Enter target IP to resolve: ");
    scanf("%s", target);

    arp_request(sender, target);

    return 0;
}
```

```
Enter sender IP: 192.168.1.2
Enter target IP to resolve: 192.168.1.3

192.168.1.2 sends ARP request: Who has 192.168.1.3?
192.168.1.3 replies: 192.168.1.3 is at AA:BB:CC:DD:EE:03


=== Code Execution Successful ===
```



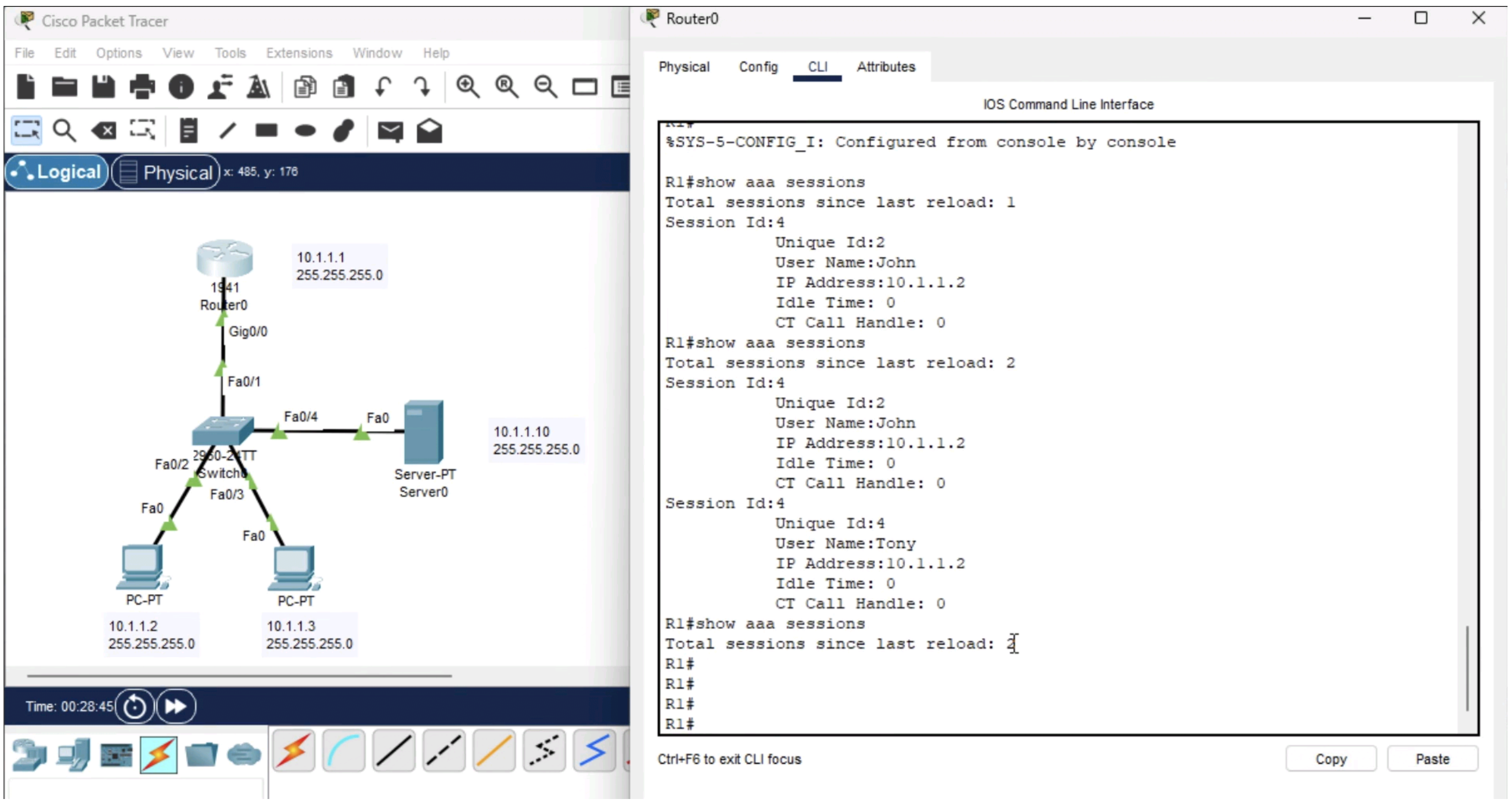
```c
        } else {
            count1 = 0;
        }
    }
    out[k] = '\0';
}

int main(void) {
    // Example: data and generator (polynomial) as bit strings
    char data[MAX] = "1101011011";    // message
    char gen[MAX]  = "10011";         // generator polynomial (CRC-4)

    char rem[MAX], codeword[MAX], stuffed[MAX], unstuffed[MAX];

    compute_crc(data, gen, rem, codeword);
    bit_stuff(codeword, stuffed);
    bit_unstuff(stuffed, unstuffed); // sanity check

    printf("Data             : %s\n", data);
    printf("Generator (G)    : %s\n", gen);
    printf("CRC remainder    : %s\n", rem);
    printf("Codeword         : %s\n", codeword);
    printf("Bit-stuffed Tx : %s\n", stuffed);
    printf("Unstuffed        : %s\n", unstuffed);

    return 0;
}
```

```
Data            : 1101011011
Generator (G)   : 10011
CRC remainder   : 1110
Codeword        : 11010110111110
Bit-stuffed Tx  : 110101101111100
Unstuffed       : 11010110111110


=== Code Execution Successful ===
```

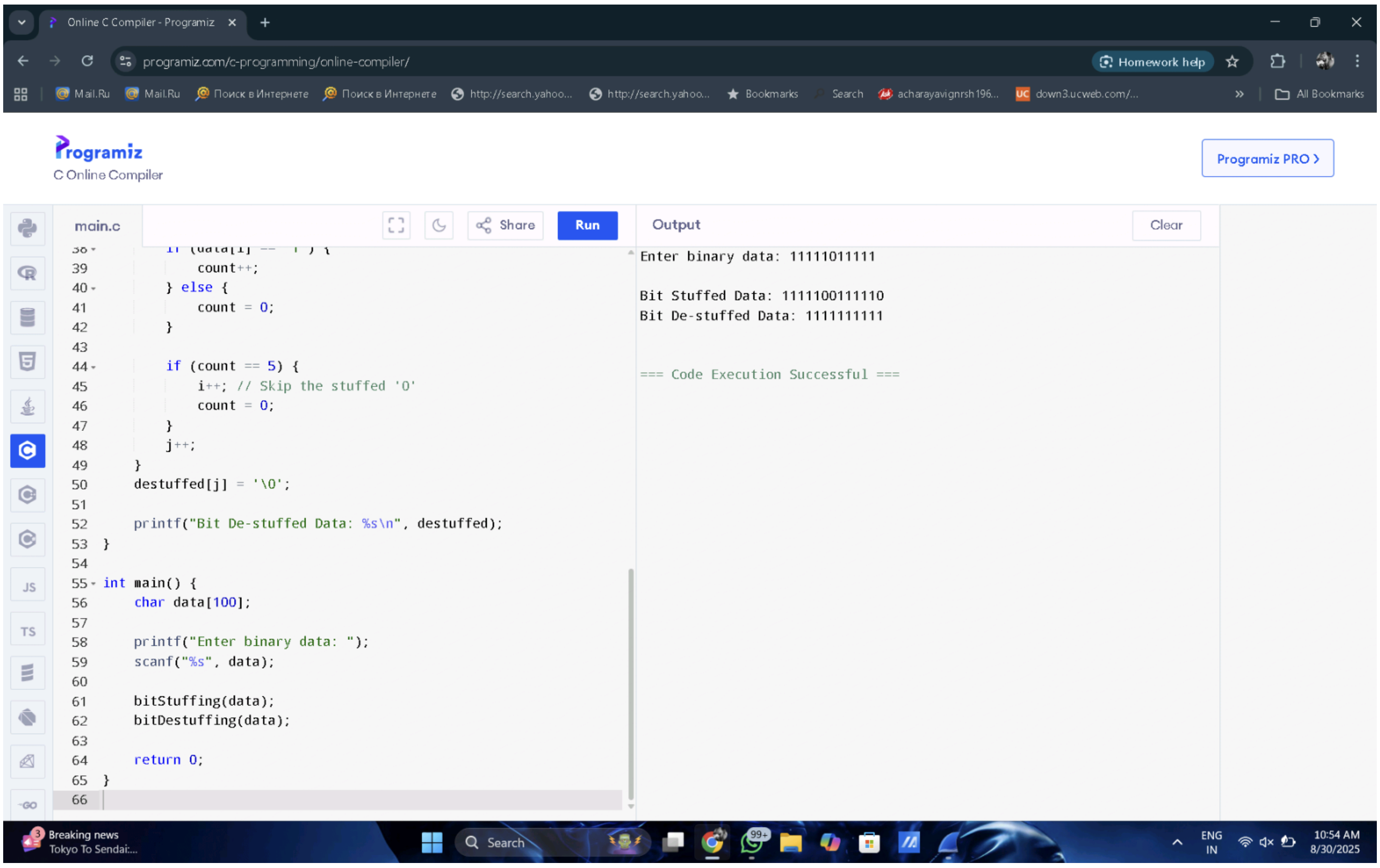


```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/wait.h>

#define PORT 5353
#define BUF_SIZE 100

// Simple DNS lookup function
const char* dns_lookup(const char* domain) {
    if (strcmp(domain, "example.com") == 0)
        return "93.184.216.34";
    else if (strcmp(domain, "localhost") == 0)
        return "127.0.0.1";
    else
        return "0.0.0.0"; // not found
}

```

```
DNS Server listening on UDP port 5353...
Enter domain name to resolve: example.com
Server received query: example.com
Server sent IP: 93.184.216.34
Client received IP: 93.184.216.34


=== Code Execution Successful ===
```

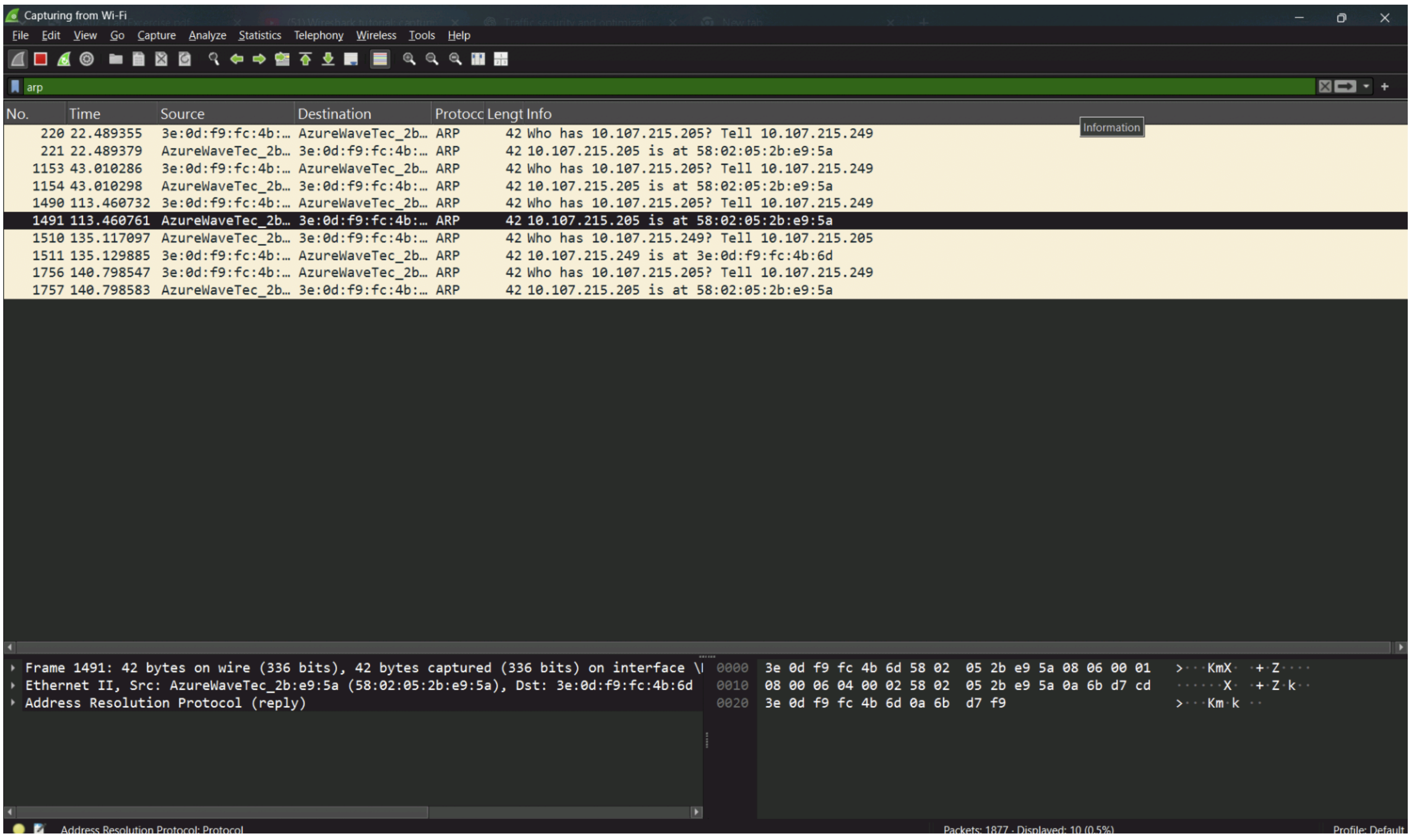


```
%SYS-5-CONFIG_I: Configured from console by console

R1#show aaa sessions
Total sessions since last reload: 1
Session Id:4
           Unique Id:2
           User Name:John
           IP Address:10.1.1.2
           Idle Time: 0
           CT Call Handle: 0
R1#show aaa sessions
Total sessions since last reload: 2
Session Id:4
           Unique Id:2
           User Name:John
           IP Address:10.1.1.2
           Idle Time: 0
           CT Call Handle: 0
Session Id:4
           Unique Id:4
           User Name:Tony
           IP Address:10.1.1.2
           Idle Time: 0
           CT Call Handle: 0
R1#show aaa sessions
Total sessions since last reload: 2
R1#
R1#
R1#
R1#
```


Online C Compiler - Programiz
programiz.com/c-programming/online-compiler/
Programiz
C Online Compiler
Programiz PRO
main.c
Share
Run
Output
Clear

```c
        if (data[i] == '1') {
            count++;
        } else {
            count = 0;
        }

        if (count == 5) {
            i++; // Skip the stuffed '0'
            count = 0;
        }
        j++;
    }
    destuffed[j] = '\0';

    printf("Bit De-stuffed Data: %s\n", destuffed);
}

int main() {
    char data[100];

    printf("Enter binary data: ");
    scanf("%s", data);

    bitStuffing(data);
    bitDestuffing(data);

    return 0;
}
```

```
Enter binary data: 11111011111

Bit Stuffed Data: 1111100111110
Bit De-stuffed Data: 1111111111


=== Code Execution Successful ===
```


Capturing from Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
arp
Information

```
No.   Time       Source            Destination       Protocc Lengt Info
  220 22.489355  3e:0d:f9:fc:4b:…  AzureWaveTec_2b…  ARP        42 Who has 10.107.215.205? Tell 10.107.215.249
  221 22.489379  AzureWaveTec_2b…  3e:0d:f9:fc:4b:…  ARP        42 10.107.215.205 is at 58:02:05:2b:e9:5a
 1153 43.010286  3e:0d:f9:fc:4b:…  AzureWaveTec_2b…  ARP        42 Who has 10.107.215.205? Tell 10.107.215.249
 1154 43.010298  AzureWaveTec_2b…  3e:0d:f9:fc:4b:…  ARP        42 10.107.215.205 is at 58:02:05:2b:e9:5a
 1490 113.460732 3e:0d:f9:fc:4b:…  AzureWaveTec_2b…  ARP        42 Who has 10.107.215.205? Tell 10.107.215.249
 1491 113.460761 AzureWaveTec_2b…  3e:0d:f9:fc:4b:…  ARP        42 10.107.215.205 is at 58:02:05:2b:e9:5a
 1510 135.117097 AzureWaveTec_2b…  3e:0d:f9:fc:4b:…  ARP        42 Who has 10.107.215.249? Tell 10.107.215.205
 1511 135.129885 3e:0d:f9:fc:4b:…  AzureWaveTec_2b…  ARP        42 10.107.215.249 is at 3e:0d:f9:fc:4b:6d
 1756 140.798547 3e:0d:f9:fc:4b:…  AzureWaveTec_2b…  ARP        42 Who has 10.107.215.205? Tell 10.107.215.249
 1757 140.798583 AzureWaveTec_2b…  3e:0d:f9:fc:4b:…  ARP        42 10.107.215.205 is at 58:02:05:2b:e9:5a
```

```
› Frame 1491: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface \I
› Ethernet II, Src: AzureWaveTec_2b:e9:5a (58:02:05:2b:e9:5a), Dst: 3e:0d:f9:fc:4b:6d
› Address Resolution Protocol (reply)
```

```
0000  3e 0d f9 fc 4b 6d 58 02  05 2b e9 5a 08 06 00 01   >···KmX· ·+·Z····
0010  08 00 06 04 00 02 58 02  05 2b e9 5a 0a 6b d7 cd   ······X· ·+·Z·k··
0020  3e 0d f9 fc 4b 6d 0a 6b  d7 f9                     >···Km·k ··
```

```
Address Resolution Protocol: Protocol          Packets: 1877 · Displayed: 10 (0.5%)          Profile: Default
```