PROGRAM CODE

server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>

void main() {
	int server_fd, client_fd;
	struct sockaddr_in address;
	int PORT, addrlen = sizeof(address);

	printf("Enter port: ");
	scanf("%d", &PORT);

	if((server_fd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
		printf("Socket creation failed!\n");
		exit(1);
	}

	address.sin_family = AF_INET;
	address.sin_addr.s_addr = INADDR_ANY;
	address.sin_port = htons(PORT);

	if(bind(server_fd, (struct sockaddr*) &address, addrlen) < 0) {
		printf("Socket binding failed!\n");
		exit(1);
	}

	if(listen(server_fd, 1) < 0) {
		printf("Listening failed!\n");
		exit(1);
	}

	if((client_fd = accept(server_fd, (struct sockaddr*) &address,
(socklen_t*) &addrlen)) < 0) {
		printf("Connection failed!\n");
		exit(1);
	}

	int n;

	while(1) {
		int k = recv(client_fd, &n, sizeof(int), 0);

		if(k < 0) {
			printf("Receive failed!\n");
			break;
		} else if(k == 0) {
			printf("Receive complete!\n");
			break;
		} else {
			printf("Received: %d\n", n);
		}
	}

	close(server_fd);
	close(client_fd);
}
```

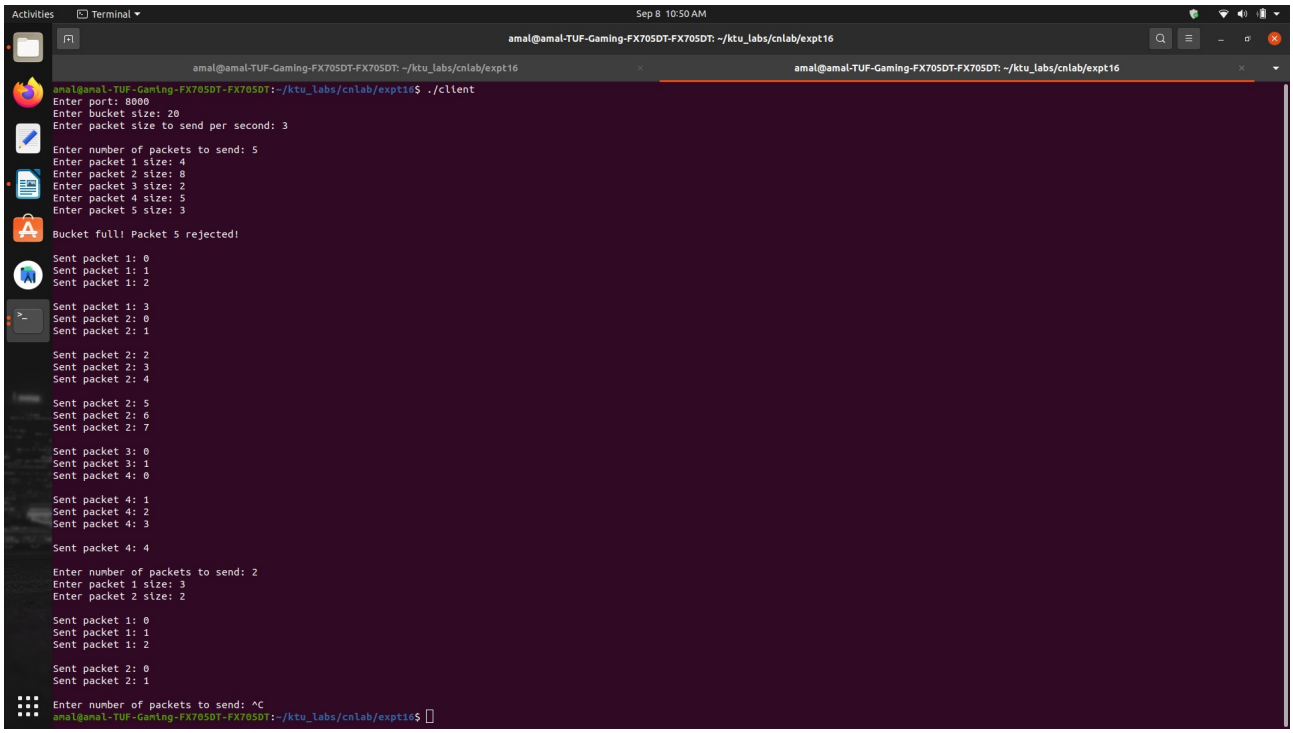
client.c

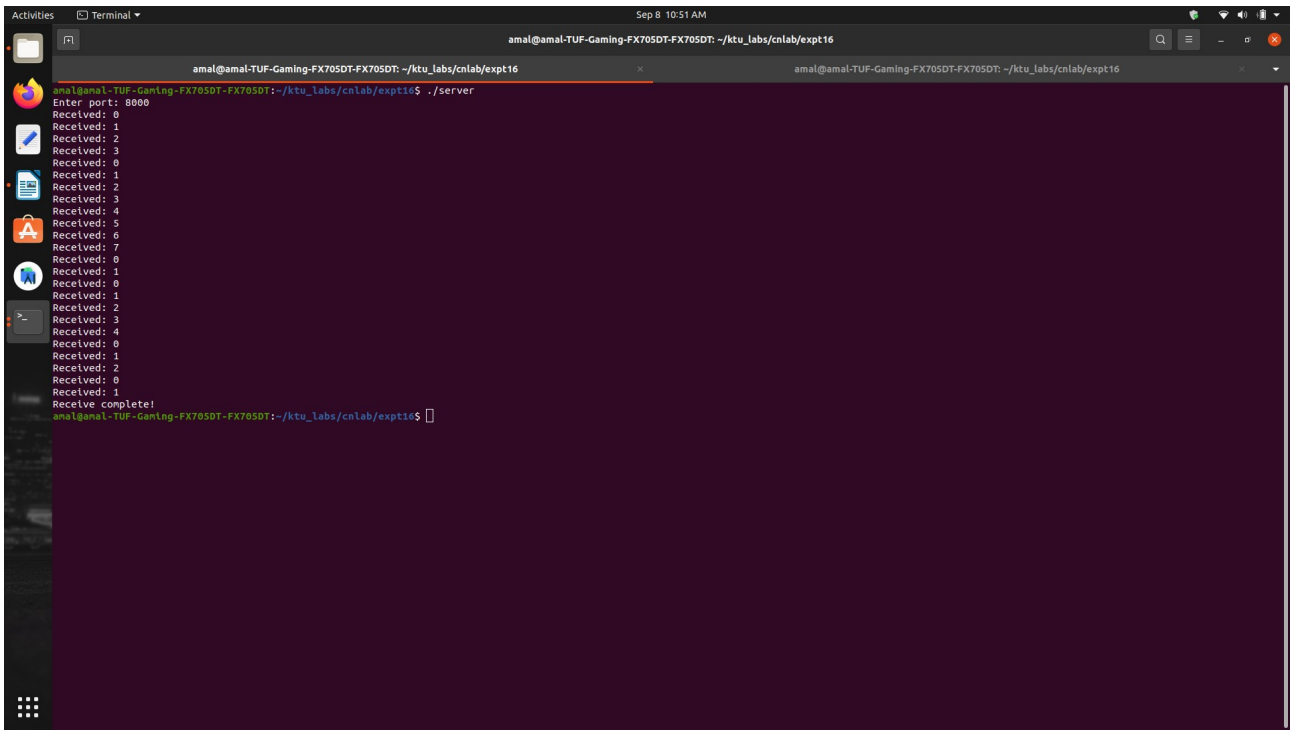
```
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <unistd.h>

typedef struct queue {
	int* arr;
	int FRONT;
	int REAR;
	int SIZE;
	int REMAINING;
} queue;

void initQueue(queue* q, int size) {
	q->FRONT = -1;
	q->REAR = -1;
	q->SIZE = size;
	q->REMAINING = size;
	q->arr = malloc(size * sizeof(int));
}

void enqueue(queue* q, int i) {
	if(q->FRONT == -1)
		q->FRONT = 0;
	else if((q->REAR + 1) % q->SIZE == q->FRONT)
		return;

	q->REAR = (q->REAR + 1) % q->SIZE;
	q->arr[q->REAR] = i;
}

int dequeue(queue* q) {
	int data;

	if(q->FRONT == -1)
		return -1;
	else if(q->FRONT == q->REAR) {
		data = q->arr[q->FRONT];
		q->FRONT = -1;
		q->REAR = -1;
	} else {
		data = q->arr[q->FRONT];
		q->FRONT = (q->FRONT + 1) % q->SIZE;
	}

	q->REMAINING++;
	return data;
}

void sendData(int client_fd, queue* q, int size) {
	int k = 0, n;

	while(1) {
		printf("\n");
		for(int i = 0; i < size; i++) {
			n = dequeue(q);

			if(n == -1)
				return;
```

```
                else if(n == 0)
                        k++;

                if(send(client_fd, &n, sizeof(int), 0) < 0) {
                        printf("Send failed!\n");
                        exit(1);
                } else {
                        printf("Sent packet %d: %d\n", k, n);
                }
        }

        sleep(1);
    }
}

void main(int argc, char* argv[]) {
    int PORT, client_fd;
    struct sockaddr_in serv_addr;

    printf("Enter port: ");
    scanf("%d", &PORT);

    client_fd = socket(AF_INET, SOCK_STREAM, 0);

    if(client_fd < 0) {
        printf("Socket creation failed!\n");
        exit(1);
    }

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = INADDR_ANY;
    serv_addr.sin_port = htons(PORT);

    if(connect(client_fd, (struct sockaddr*) &serv_addr, sizeof(serv_addr)) <
0) {
        printf("Connection failed!\n");
        exit(1);
    }

    queue q;
    int num, size;

    printf("Enter bucket size: ");
    scanf("%d", &size);

    initQueue(&q, size);

    printf("Enter packet size to send per second: ");
    scanf("%d", &size);

    while(1) {
        printf("\nEnter number of packets to send: ");
        scanf("%d", &num);

        int packets[num];

        for(int i = 0; i < num; i++) {
            printf("Enter packet %d size: ", i + 1);
            scanf("%d", &packets[i]);
        }

        for(int i = 0; i < num; i++) {
            if(q.REMAINING < packets[i]) {
                printf("\nBucket full! Packet %d rejected!\n", i + 1);
```

```
                        continue;
                }

                for(int j = 0; j < packets[i]; j++) {
                        enqueue(&q, j);
                        q.REMAINING--;
                }
        }

        sendData(client_fd, &q, size);
    }

    close(client_fd);
}
```

OUTPUT


Activities
Terminal
Sep 8 10:50 AM
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt16
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt16$ ./cllent
Enter port: 8000
Enter bucket size: 20
Enter packet size to send per second: 3
Enter number of packets to send: 5
Enter packet 1 size: 4
Enter packet 2 size: 8
Enter packet 3 size: 2
Enter packet 4 size: 5
Enter packet 5 size: 3
Bucket full! Packet 5 rejected!
Sent packet 1: 0
Sent packet 1: 1
Sent packet 1: 2
Sent packet 1: 3
Sent packet 2: 0
Sent packet 2: 1
Sent packet 2: 2
Sent packet 2: 3
Sent packet 2: 4
Sent packet 2: 5
Sent packet 2: 6
Sent packet 2: 7
Sent packet 3: 0
Sent packet 3: 1
Sent packet 4: 0
Sent packet 4: 1
Sent packet 4: 2
Sent packet 4: 3
Sent packet 4: 4
Enter number of packets to send: 2
Enter packet 1 size: 3
Enter packet 2 size: 2
Sent packet 1: 0
Sent packet 1: 1
Sent packet 1: 2
Sent packet 2: 0
Sent packet 2: 1
Enter number of packets to send: ^C
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt16$


Activities
Terminal
Sep 8 10:51 AM
amal@amal-TUF-Gaming-FX705DT-FX705DT: ~/ktu_labs/cnlab/expt16
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt16$ ./server
Enter port: 8000
Received: 0
Received: 1
Received: 2
Received: 3
Received: 0
Received: 1
Received: 2
Received: 3
Received: 4
Received: 5
Received: 6
Received: 7
Received: 0
Received: 1
Received: 0
Received: 1
Received: 2
Received: 3
Received: 4
Received: 0
Received: 1
Received: 2
Received: 0
Received: 1
Receive complete!
amal@amal-TUF-Gaming-FX705DT-FX705DT:~/ktu_labs/cnlab/expt16$