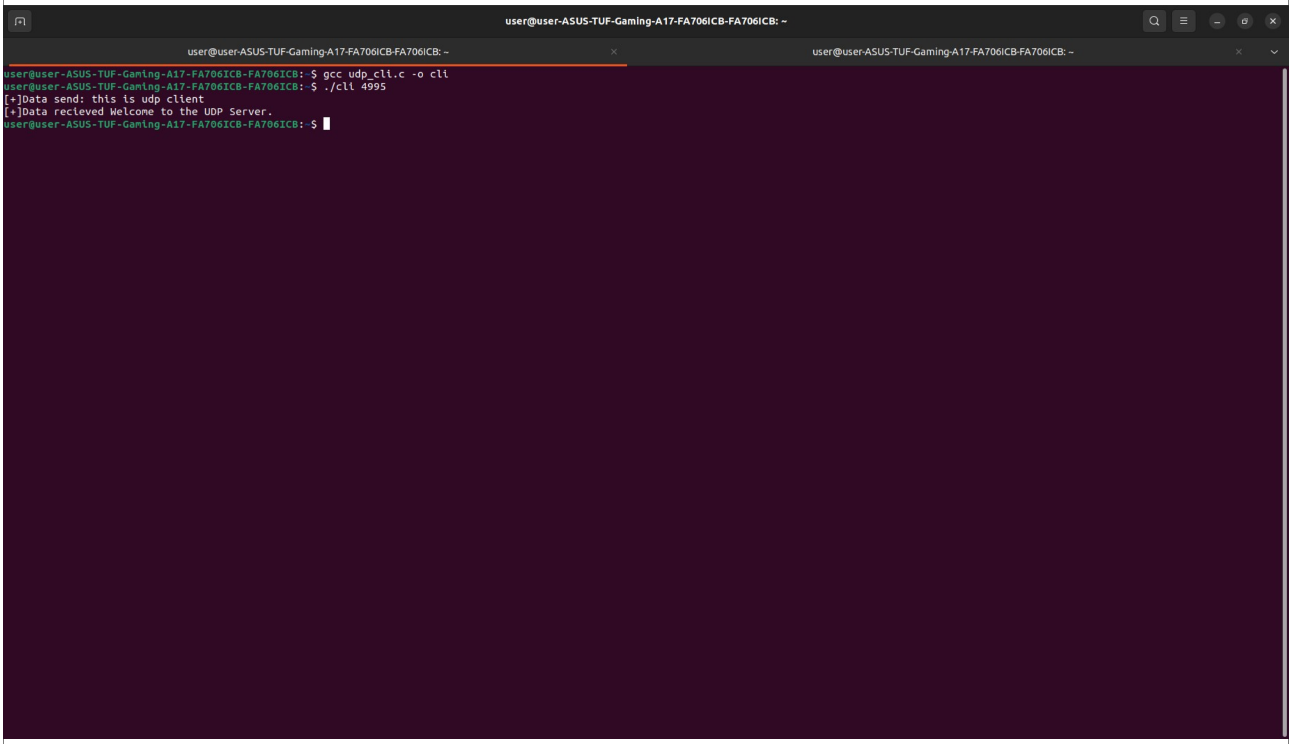
Q. UDP mathematical expression solved.

udp_cli.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>

int main(int argc, char **argv){

  if (argc != 2) {
    printf("Usage: %s <port>\n", argv[0]);
    exit(0);
  }

  char *ip = "127.0.0.1";
  int port = atoi(argv[1]);

  int sockfd;
  struct sockaddr_in addr;
  char buffer[1024];
  char buf[1024];
  int t,i;
  socklen_t addr_size;

  sockfd = socket(AF_INET, SOCK_DGRAM, 0);
  memset(&addr, '\0', sizeof(addr));
  addr.sin_family = AF_INET;
  addr.sin_port = htons(port);
  addr.sin_addr.s_addr = inet_addr(ip);

  bzero(buffer, 1024);
  strcpy(buffer, "this is udp client");
  sendto(sockfd, buffer, 1024, 0, (struct sockaddr*)&addr, sizeof(addr));
  printf("[+]Data send: %s\n", buffer);

  bzero(buffer, 1024);
  addr_size = sizeof(addr);
  recvfrom(sockfd, buf, 1024, 0, (struct sockaddr*)&addr, &addr_size);
  printf("[+]Data recieved %s\n", buf);
/*  for(i=0;i<10;i++)
  {
  printf("%d", buf[i]);

  }*/

  return 0;
}
```

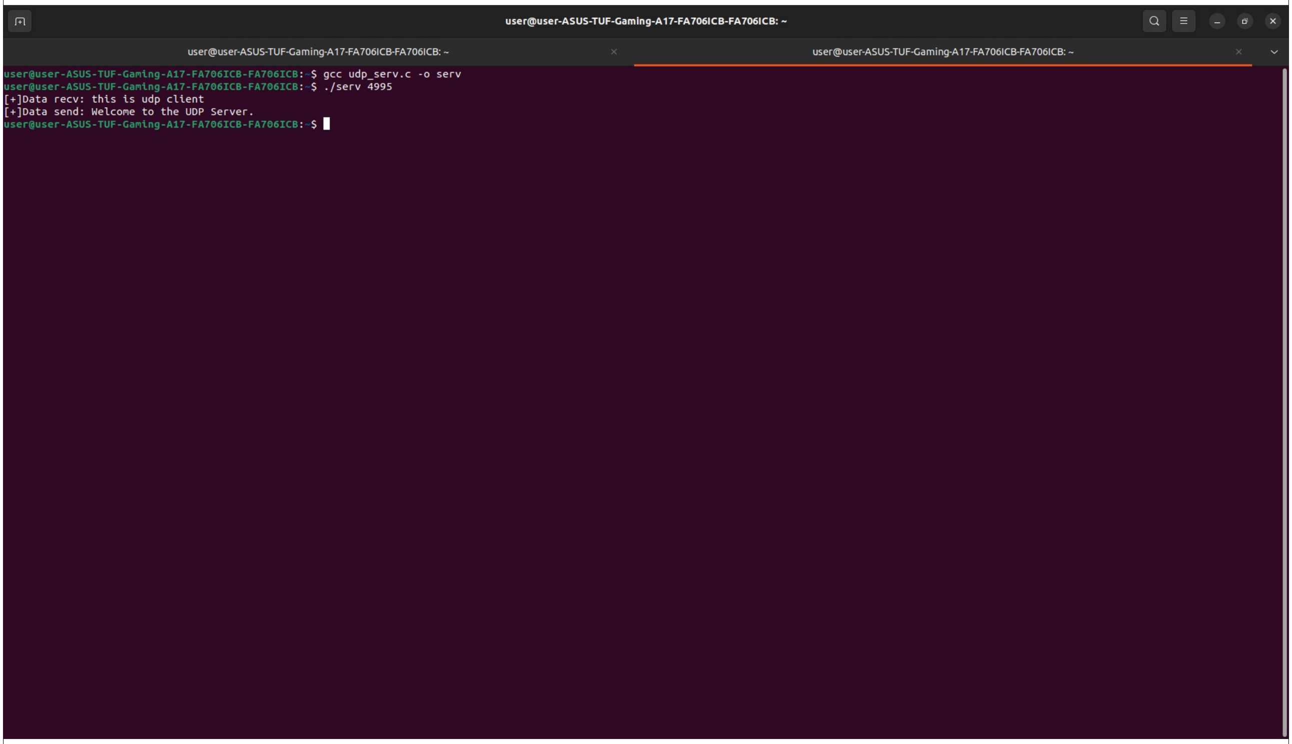
```
udp_serv.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
int main(int argc, char **argv){
 char buffer[1024];
  if (argc != 2) {
    printf("Usage: %s <port>\n", argv[0]);
    exit(0);
  }

  char *ip = "127.0.0.1";
  int port = atoi(argv[1]);

  int sockfd;
  struct sockaddr_in server_addr, client_addr;

  socklen_t addr_size;
  int n;

  sockfd = socket(AF_INET, SOCK_DGRAM, 0);
  if (sockfd < 0) {
    perror("[-]socket error");
    exit(1);
  }

  memset(&server_addr, '\0', sizeof(server_addr));
  server_addr.sin_family = AF_INET;
  server_addr.sin_port = htons(port);
  server_addr.sin_addr.s_addr = inet_addr(ip);

  n = bind(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
  if (n < 0){
    perror("[-]bind error");
    exit(1);
  }

  bzero(buffer, 1024);
  addr_size = sizeof(client_addr);
  recvfrom(sockfd, buffer, 1024, 0, (struct sockaddr*)&client_addr, &addr_size);
  printf("[+]Data recv: %s\n", buffer);
 // int res=addsub();



  bzero(buffer, 1024);
  strcpy(buffer, "Welcome to the UDP Server.");
  sendto(sockfd, buffer, 1024, 0, (struct sockaddr*)&client_addr, sizeof(client_addr));
  printf("[+]Data send: %s\n", buffer);

  return 0;
}
```

OUTPUT:-



user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~

```
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$ gcc udp_cli.c -o cli
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$ ./cli 4995
[+]Data send: this is udp client
[+]Data recieved Welcome to the UDP Server.
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$ 
```



user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB: ~

```
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$ gcc udp_serv.c -o serv
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$ ./serv 4995
[+]Data recv: this is udp client
[+]Data send: Welcome to the UDP Server.
user@user-ASUS-TUF-Gaming-A17-FA706ICB-FA706ICB:~$ 
```

Q. UDP mathematical expression solved.

cli.c

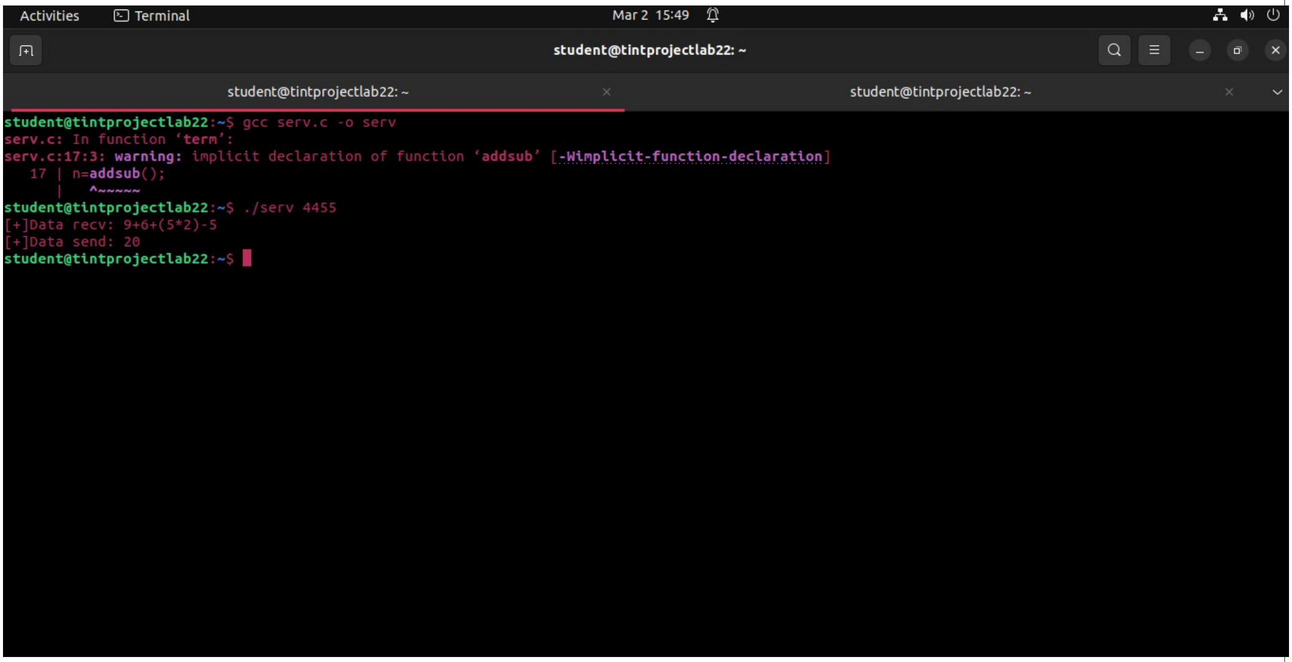
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>

int main(int argc, char **argv){

  if (argc != 2) {
    printf("Usage: %s <port>\n", argv[0]);
    exit(0);
  }

  char *ip = "127.0.0.1";
  int port = atoi(argv[1]);

  int sockfd;
  struct sockaddr_in addr;
  char buffer[1024];
  int buf[10];
  int t,i;
  socklen_t addr_size;

  sockfd = socket(AF_INET, SOCK_DGRAM, 0);
  memset(&addr, '\0', sizeof(addr));
  addr.sin_family = AF_INET;
  addr.sin_port = htons(port);
  addr.sin_addr.s_addr = inet_addr(ip);

  bzero(buffer, 1024);
  strcpy(buffer, "9+6+(5*2)-5");
  sendto(sockfd, buffer, 1024, 0, (struct sockaddr*)&addr, sizeof(addr));
  printf("[+]Data send: %s\n", buffer);

  bzero(buffer, 1024);
  addr_size = sizeof(addr);
  recvfrom(sockfd, buf, 10, 0, (struct sockaddr*)&addr, &addr_size);
/*  for(i=0;i<10;i++)
  {
  printf("%d", buf[i]);

  }*/

  return 0;
}
```

serv.c

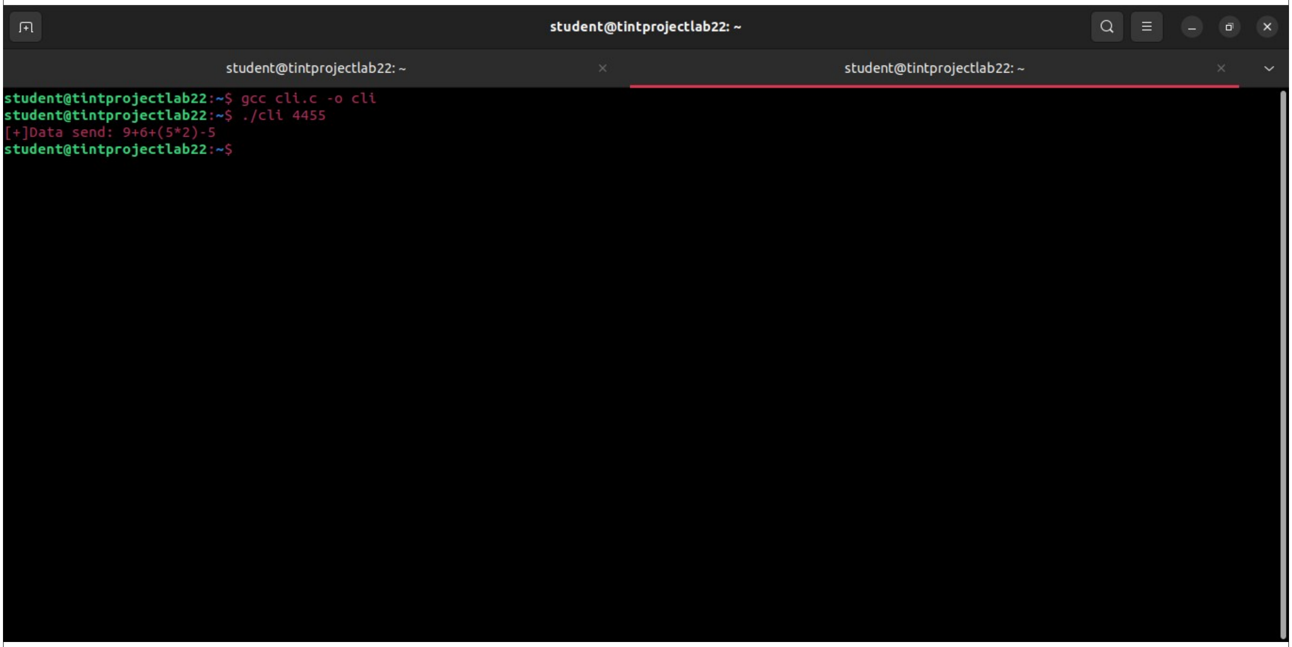
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
char buffer[1024];
int pos=0;
int term()
{
int n=0;

if(buffer[pos]=='(')
{
pos++;
n=addsub();
if(buffer[pos]==')')
{
pos++;
return n;
}
}
else
{
while('0'<=buffer[pos] && buffer[pos]<='9'){
n=n*10+(buffer[pos] - '0');
pos++;
}
}
return n;


}
int muldiv()
{
int first,second;
first=term();
for(;;)
{
if(buffer[pos]=='*')
{
pos++;
second=term();
first=first*second;
}
else if(buffer[pos]=='/')
{
```

```
pos++;
second=term();
first=first/second;
}
else
{
return first;
}
}
}
int addsub()
{
int first,second;
first=muldiv();
for(;;)
{
if(buffer[pos]=='+')
{
pos++;
second=muldiv();
first=first+second;
}
else if(buffer[pos]=='-')
{
pos++;
second=muldiv();
first=first-second;
}
else
{
return first;
}
}
}

int main(int argc, char **argv){

  if (argc != 2) {
    printf("Usage: %s <port>\n", argv[0]);
    exit(0);
  }

  char *ip = "127.0.0.1";
  int port = atoi(argv[1]);

  int sockfd;
  struct sockaddr_in server_addr, client_addr;

  socklen_t addr_size;
  int n;

  sockfd = socket(AF_INET, SOCK_DGRAM, 0);
  if (sockfd < 0) {
    perror("[-]socket error");
    exit(1);
  }

  memset(&server_addr, '\0', sizeof(server_addr));
  server_addr.sin_family = AF_INET;
  server_addr.sin_port = htons(port);
  server_addr.sin_addr.s_addr = inet_addr(ip);

  n = bind(sockfd, (struct sockaddr*)&server_addr, sizeof(server_addr));
  if (n < 0){
    perror("[-]bind error");
    exit(1);
  }

  bzero(buffer, 1024);
  addr_size = sizeof(client_addr);
  recvfrom(sockfd, buffer, 1024, 0, (struct sockaddr*)&client_addr, &addr_size);
  printf("[+]Data recv: %s\n", buffer);
  int res=addsub();


  bzero(buffer, 1024);
  strcpy(buffer, "Welcome to the UDP Server.");
  sendto(sockfd, buffer, 1024, 0, (struct sockaddr*)&client_addr, sizeof(client_addr));
  printf("[+]Data send: %d\n", res);

  return 0;
}
```

output:


Activities
Terminal
Mar 2 15:49
student@tintprojectlab22: ~
student@tintprojectlab22: ~
student@tintprojectlab22: ~

```
student@tintprojectlab22:~$ gcc serv.c -o serv
serv.c: In function ‘term’:
serv.c:17:3: warning: implicit declaration of function ‘addsub’ [-Wimplicit-function-declaration]
   17 | n=addsub();
      |   ^~~~~~
student@tintprojectlab22:~$ ./serv 4455
[+]Data recv: 9+6+(5*2)-5
[+]Data send: 20
student@tintprojectlab22:~$ 
```



student@tintprojectlab22: ~
student@tintprojectlab22: ~
student@tintprojectlab22: ~

```
student@tintprojectlab22:~$ gcc cli.c -o cli
student@tintprojectlab22:~$ ./cli 4455
[+]Data send: 9+6+(5*2)-5
student@tintprojectlab22:~$
```