SOCKET PROGRAMMING QUESTIONS

1.Write a C program to find the factorial of a number using TCP socket?

serTCPfact.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include <unistd.h>
#define port 5000

int fact(int n){

   if(n==0){
         return 1;
   }
   return n*fact(n-1);
}



int main()
{
   struct sockaddr_in serveraddr,newaddr;
   int sersocket,newsocket,s,size;

   int n;

   sersocket=socket(PF_INET,SOCK_STREAM,0);

   if(sersocket>0)
         printf("\nserver socket is created");


   serveraddr.sin_family= PF_INET;
   serveraddr.sin_port= htons(port);
   serveraddr.sin_addr.s_addr=htonl(INADDR_ANY);

   s=bind(sersocket,(struct sockaddr *)&serveraddr,sizeof(serveraddr));
   if(s==0)
         printf("\nbind success");

   listen(sersocket,1);

   size=sizeof(newaddr);
```

```
    printf("\nserver ready");

    newsocket=accept(sersocket,(struct sockaddr *)&newaddr,&size);

    if(newsocket>0)
        printf("\naccepted");


    recv(newsocket,&n,sizeof(n),0);
    printf("\ndata received is %d\n",n);

    int x;

    x=fact(n);

    send(newsocket,&x,sizeof(x),0);

    close(sersocket);
}
```

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o ser serTCPfact.c
alwin@debian:~/Documents/socket_programs$ ./ser

server socket is created
bind success
server ready
accepted
data received is 5
```

cliTCPfact.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include <unistd.h>
#define port 5000
void main()
{
    struct sockaddr_in serveraddr;
    int clisocket;

    int n;

    clisocket=socket(PF_INET,SOCK_STREAM,0);
```

```
    if(clisocket>0)
            printf("\nclient socket created");


    serveraddr.sin_family= PF_INET;
    serveraddr.sin_port= htons(port);
    serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");

    connect(clisocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));

    printf("\nEnter the number:");
    scanf("%d",&n);

    send(clisocket,&n,sizeof(n),0);

    int fact=1;
    recv(clisocket,&fact,sizeof(fact),0);

    printf("Factorial of %d is:%d\n",n,fact);

    close(clisocket);
}
```

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o cli cliTCPfact.c
alwin@debian:~/Documents/socket_programs$ ./cli

client socket created
Enter the number:5
Factorial of 5 is:120
```

2.Write a C program to find the factorial of a number using UDP socket?

serUDPfact.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include <unistd.h>
#define port 4000

int fact(int n){
```

```
    if(n==0){
        return 1;
    }
    return n*fact(n-1);
}


void main()
{
    struct sockaddr_in serveraddr,newaddr;
    int sersocket,s,size;


    sersocket=socket(AF_INET,SOCK_DGRAM,0);

    if(sersocket>0)
        printf("\nServer socket created");

    serveraddr.sin_family=AF_INET;
    serveraddr.sin_port=htons(port);
    serveraddr.sin_addr.s_addr=htonl(INADDR_ANY);


    s=bind(sersocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
    if(s==0)
        printf("\nBind success");

    size=sizeof(newaddr);

    int n=0;

    recvfrom(sersocket,&n,sizeof(n),0,(struct sockaddr*)&newaddr,&size);

    printf("\nData recieved:%d\n",n);

    int x;

    x=fact(n);

    sendto(sersocket,&x,sizeof(x),0,(struct sockaddr*)&newaddr,sizeof(newaddr));

    close(sersocket);
}
```

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o ser serUDPfact.c
alwin@debian:~/Documents/socket_programs$ ./ser

Server socket created
Bind success
Data recieved:5
```

cliUDPfact.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include <unistd.h>
#define port 4000
void main()
{
   struct sockaddr_in serveraddr,newaddr;
   int clisocket,size;


   clisocket=socket(AF_INET,SOCK_DGRAM,0);

   if(clisocket>0)
         printf("\nClient socket created");


   serveraddr.sin_family=AF_INET;
   serveraddr.sin_port=htons(port);
   serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");

   int n;

   printf("\nEnter the number:");
   scanf("%d",&n);

   sendto(clisocket,&n,sizeof(n),0,(struct sockaddr*)&serveraddr,sizeof(serveraddr));

   size=sizeof(newaddr);

   int fact=1;

   recvfrom(clisocket,&fact,sizeof(fact),0,(struct sockaddr*)&newaddr,&size);

   printf("Factorial of %d is:%d\n",n,fact);

   close(clisocket);
}
```

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o cli cliUDPfact.c
alwin@debian:~/Documents/socket_programs$ ./cli

Client socket created
Enter the number:5
Factorial of 5 is:120
```

3. Write a C program to check a number is prime or not using TCP socket?

serTCPprime.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include <unistd.h>
#define port 5000

int prime(int n){
   if(n<=1)
   {
         return 0;
   }
   for(int i=2;i<n/2;i++)
   {
         if (n%i==0)
         {
                return 0;
         }
   }
   return 1;
}



int main()
{
   struct sockaddr_in serveraddr,newaddr;
   int sersocket,newsocket,s,size;

   int n;

   sersocket=socket(PF_INET,SOCK_STREAM,0);
```

```
    if(sersocket>0)
            printf("\nserver socket is created");


    serveraddr.sin_family= PF_INET;
    serveraddr.sin_port= htons(port);
    serveraddr.sin_addr.s_addr=htonl(INADDR_ANY);

    s=bind(sersocket,(struct sockaddr *)&serveraddr,sizeof(serveraddr));
    if(s==0)
            printf("\nbind success");

    listen(sersocket,1);

    size=sizeof(newaddr);

    printf("\nserver ready");

    newsocket=accept(sersocket,(struct sockaddr *)&newaddr,&size);

    if(newsocket>0)
            printf("\naccepted");


    recv(newsocket,&n,sizeof(n),0);
    printf("\ndata received is %d\n",n);

    char buffer[100];

    if(prime(n)==0)
    {
            sprintf(buffer,"not a prime");
    }
    else
    {
            sprintf(buffer,"prime");
    }

    send(newsocket,buffer,sizeof(buffer),0);

    close(sersocket);
}
```

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o ser serTCPprime.c
alwin@debian:~/Documents/socket_programs$ ./ser

server socket is created
bind success
server ready
accepted
data received is 10
```

cliTCPprime.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include <unistd.h>
#define port 5000
void main()
{
   struct sockaddr_in serveraddr;
   int clisocket;

   int n;

   clisocket=socket(PF_INET,SOCK_STREAM,0);

   if(clisocket>0)
         printf("\nclient socket created");


   serveraddr.sin_family= PF_INET;
   serveraddr.sin_port= htons(port);
   serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");

   connect(clisocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));

   printf("\nEnter the number:");
   scanf("%d",&n);

   send(clisocket,&n,sizeof(n),0);
```

```
    char buffer[100];
    recv(clisocket,buffer,sizeof(buffer),0);

    printf("%d is a %s number\n",n,buffer);

    close(clisocket);
}
```

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o cli cliTCPprime.c
alwin@debian:~/Documents/socket_programs$ ./cli

client socket created
Enter the number:10
10 is a not a prime number
```

4. Write a C program to check a number is prime or not using UDP socket?

serUDPprime.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include <unistd.h>
#define port 4000

int prime(int n){
    if(n<=1)
    {
        return 0;
    }
    for(int i=2;i<=n/2;i++)
    {
        if (n%i==0)
        {
            return 0;
        }
    }
    return 1;
}
```

```
void main()
{
   struct sockaddr_in serveraddr,newaddr;
   int sersocket,s,size;


   sersocket=socket(AF_INET,SOCK_DGRAM,0);

   if(sersocket>0)
         printf("\nServer socket created");

   serveraddr.sin_family=AF_INET;
   serveraddr.sin_port=htons(port);
   serveraddr.sin_addr.s_addr=htonl(INADDR_ANY);


   s=bind(sersocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));
   if(s==0)
         printf("\nBind success");

   size=sizeof(newaddr);

   int n=0;

   recvfrom(sersocket,&n,sizeof(n),0,(struct sockaddr*)&newaddr,&size);

   printf("\nData recieved:%d\n",n);

   char buffer[100];

   if(prime(n)==0)
   {
         sprintf(buffer,"not a prime");
   }
   else
   {
         sprintf(buffer,"prime");
   }

   sendto(sersocket,buffer,sizeof(buffer),0,(struct sockaddr*)&newaddr,sizeof(newaddr));

   close(sersocket);
}
```

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o ser serUDPprime.c
alwin@debian:~/Documents/socket_programs$ ./ser

Server socket created
Bind success
Data recieved:10
```

cliUDPprime.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include <unistd.h>
#define port 4000
void main()
{
   struct sockaddr_in serveraddr,newaddr;
   int clisocket,size;


   clisocket=socket(AF_INET,SOCK_DGRAM,0);

   if(clisocket>0)
         printf("\nClient socket created");


   serveraddr.sin_family=AF_INET;
   serveraddr.sin_port=htons(port);
   serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");

   int n;

   printf("\nEnter the number:");
   scanf("%d",&n);

   sendto(clisocket,&n,sizeof(n),0,(struct sockaddr*)&serveraddr,sizeof(serveraddr));

   size=sizeof(newaddr);

   char buffer[100];
```

```
    recvfrom(clisocket,buffer,sizeof(buffer),0,(struct sockaddr*)&newaddr,&size);

    printf("%d is %s",n,buffer);
    close(clisocket);
}
```

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o cli cliUDPprime.c
alwin@debian:~/Documents/socket_programs$ ./cli

Client socket created
Enter the number:10
10 is not a prime
```

5. Write a C program to print the reverse of a string using TCP socket?

serTCPrev.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include<unistd.h>
#include<string.h>
#define port 5000

void str_reverse(char buffer[]) {

        char reverse[100];
        int length = strlen(buffer);

        for (int i = 0; i < length; i++) {
        reverse[i] = buffer[length - i - 1];
        }

        reverse[length] = '\0';
        strcpy(buffer,reverse);
}


int main()
{
```

```
    struct sockaddr_in serveraddr,newaddr;
    int sersocket,newsocket,s,size;

    sersocket=socket(PF_INET,SOCK_STREAM,0);

    if(sersocket>0)
            printf("\nserver socket is created");


    serveraddr.sin_family= PF_INET;
    serveraddr.sin_port= htons(port);
    serveraddr.sin_addr.s_addr=htonl(INADDR_ANY);

    s=bind(sersocket,(struct sockaddr *)&serveraddr,sizeof(serveraddr));
    if(s==0)
            printf("\nbind success");

    listen(sersocket,1);

    size=sizeof(newaddr);

    printf("\nserver ready");

    newsocket=accept(sersocket,(struct sockaddr *)&newaddr,&size);

    if(newsocket>0)
            printf("\naccepted");


    char buffer[100];

    recv(newsocket,buffer,sizeof(buffer),0);
    printf("\ndata received is %s\n",buffer);

    str_reverse(buffer);

    send(newsocket,buffer,sizeof(buffer),0);

    close(sersocket);
}
```

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o ser serTCPrev.c
alwin@debian:~/Documents/socket_programs$ ./ser

server socket is created
bind success
server ready
accepted
data received is Hello
```

cliTCPrev.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include <unistd.h>
#define port 5000
void main()
{
   struct sockaddr_in serveraddr;
   int clisocket;


   clisocket=socket(PF_INET,SOCK_STREAM,0);

   if(clisocket>0)
         printf("\nclient socket created");


   serveraddr.sin_family= PF_INET;
   serveraddr.sin_port= htons(port);
   serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");

   connect(clisocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));

   char buffer[100];

   printf("\nEnter the string:");
   scanf("%s",buffer);

   send(clisocket,buffer,sizeof(buffer),0);


   recv(clisocket,buffer,sizeof(buffer),0);
```

```
    printf("The reversed string is: %s\n",buffer);

    close(clisocket);
}
```

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o cli cliTCPrev.c
alwin@debian:~/Documents/socket_programs$ ./cli

client socket created
Enter the string:Hello
The reversed string is: olleH
```

6.Write a C program to find the fibonacci series of a number using TCP socket?

serTCPfibo.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include<unistd.h>
#include<string.h>
#define port 5000

void fibo(int n,int array[]) {

        array[0]=0;
        array[1]=1;

        for(int i=2;i<=n;i++)
        {
        array[i]=array[i-1]+array[i-2];
        }

}

int main()
{
    struct sockaddr_in serveraddr,newaddr;
    int sersocket,newsocket,s,size;
```

```
sersocket=socket(PF_INET,SOCK_STREAM,0);

if(sersocket>0)
        printf("\nserver socket is created");


serveraddr.sin_family= PF_INET;
serveraddr.sin_port= htons(port);
serveraddr.sin_addr.s_addr=htonl(INADDR_ANY);

s=bind(sersocket,(struct sockaddr *)&serveraddr,sizeof(serveraddr));
if(s==0)
        printf("\nbind success");

listen(sersocket,1);

size=sizeof(newaddr);

printf("\nserver ready");

newsocket=accept(sersocket,(struct sockaddr *)&newaddr,&size);

if(newsocket>0)
        printf("\naccepted");


int n,a[100];

recv(newsocket,&n,sizeof(n),0);
printf("\ndata received is %d\n",n);

fibo(n,a);

char buffer[100];

for(int i=0;i<n;i++)
{
        sprintf(buffer+strlen(buffer),"%d",a[i]);
}

send(newsocket,buffer,sizeof(buffer),0);

close(sersocket);
```

}

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o ser serTCPfibo.c
alwin@debian:~/Documents/socket_programs$ ./ser

server socket is created
bind success
server ready
accepted
data received is 5
```

cliTCPfibo.c

```
#include<stdio.h>
#include<arpa/inet.h>
#include<unistd.h>
#include<string.h>
#define port 5000
void main()
{
   struct sockaddr_in serveraddr;
   int clisocket;


   clisocket=socket(PF_INET,SOCK_STREAM,0);

   if(clisocket>0)
         printf("\nclient socket created");


   serveraddr.sin_family= PF_INET;
   serveraddr.sin_port= htons(port);
   serveraddr.sin_addr.s_addr=inet_addr("127.0.0.1");

   connect(clisocket,(struct sockaddr*)&serveraddr,sizeof(serveraddr));

   int n;

   printf("\nEnter the number:");
   scanf("%d",&n);

   send(clisocket,&n,sizeof(n),0);
```

```
    char buffer[100];

    recv(clisocket,buffer,sizeof(buffer),0);

    printf("Fibonacci series:");
    for(int i=0;i<strlen(buffer);i++)
    {
            printf("%c ",buffer[i]);
    }

    close(clisocket);
}
```

OUTPUT:

```
alwin@debian:~/Documents/socket_programs$ gcc -o cli cliTCPfibo.c
alwin@debian:~/Documents/socket_programs$ ./cli

client socket created
Enter the number:5
Fibonacci series:0 1 1 2 3
```