/\*WAP to create a TCP socket through which client will send a string to the server , server will echo back the string to the client \*/

**CLIENT**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

#include<unistd.h>

int main()

{

int i\_534,sockfd\_534;

char buf\_534[100];

struct sockaddr\_in sa\_addr\_534;

sockfd\_534=socket(AF\_INET,SOCK\_STREAM,0);

sa\_addr\_534.sin\_family=AF\_INET;

sa\_addr\_534.sin\_addr.s\_addr=inet\_addr("127.0.0.1"); //Loop back IP address

sa\_addr\_534.sin\_port=htons(6000);

memset(sa\_addr\_534.sin\_zero, '\0', sizeof sa\_addr\_534.sin\_zero);

i\_534=connect(sockfd\_534,(struct sockaddr \*)&sa\_addr\_534,sizeof(sa\_addr\_534));

printf("\n RAj , please enter a string ");

scanf("%s",buf\_534);

while(strcmp(buf\_534,"exit")!=0)

{

send(sockfd\_534, buf\_534, 100, 0);

for(i\_534=0; i\_534 < 100; i\_534++) buf\_534[i\_534] = '\0';

recv(sockfd\_534, buf\_534, 100, 0);

printf("\n server's response =>");

printf("%s\n", buf\_534);

printf("\n RAj , please enter a string ");

scanf("%s",buf\_534);

}

close(sockfd\_534);

}

**SERVER**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

#include<unistd.h>

int main()

{

int sockfd\_534,fd1\_534, length\_534,i\_534;

char buf\_534[100];

struct sockaddr\_in sa\_addr\_534,cl\_addr\_534;

sockfd\_534=socket(AF\_INET,SOCK\_STREAM,0);

sa\_addr\_534.sin\_family=AF\_INET;

sa\_addr\_534.sin\_addr.s\_addr=INADDR\_ANY;

sa\_addr\_534.sin\_port=htons(6000);

i\_534=bind(sockfd\_534,(struct sockaddr \*)&sa\_addr\_534,sizeof(sa\_addr\_534));

printf("test %d%d\n",sockfd\_534,i\_534);

listen(sockfd\_534,5);

length\_534=sizeof(cl\_addr\_534);

fd1\_534=accept(sockfd\_534, (struct sockaddr \*) &cl\_addr\_534,&length\_534);

for(i\_534=0; i\_534 < 100; i\_534++) buf\_534[i\_534] = '\0';

recv(fd1\_534, buf\_534, 100, 0);

while(strcmp(buf\_534,"exit")!=0)

{

send(fd1\_534, buf\_534, 100, 0);

recv(fd1\_534, buf\_534, 100, 0);

}

close(fd1\_534);

}

**OUTPUT -:**

/\*WAP to create a TCP socket through which client will send an integer number to the server , server will reverse that number and send back to the client \*/

**CLIENT**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

#include<unistd.h>

int main()

{

int i\_534,sockfd\_534;

int buf\_534;

struct sockaddr\_in sa\_addr\_534;

sockfd\_534=socket(AF\_INET,SOCK\_STREAM,0);

sa\_addr\_534.sin\_family=AF\_INET;

sa\_addr\_534.sin\_addr.s\_addr=inet\_addr("127.0.0.1"); //Loop back IP address

sa\_addr\_534.sin\_port=htons(6000);

memset(sa\_addr\_534.sin\_zero, '\0', sizeof sa\_addr\_534.sin\_zero);

i\_534=connect(sockfd\_534,(struct sockaddr \*)&sa\_addr\_534,sizeof(sa\_addr\_534));

printf("RAj , Enter a number\n");

scanf("%d",&buf\_534);

send(sockfd\_534, &buf\_534, sizeof(buf\_534), 0);

printf("\n server's response ");

recv(sockfd\_534, &buf\_534, sizeof(buf\_534), 0);

printf("\n reversed number => %d\n", buf\_534);

close(sockfd\_534);

}

**SERVER**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

#include<unistd.h>

int main()

{

int sockfd\_534,fd1\_534, length\_534,i\_534;

int buf\_534,rev\_534=0;

struct sockaddr\_in sa\_addr\_534,cl\_addr\_534;

sockfd\_534=socket(AF\_INET,SOCK\_STREAM,0);

sa\_addr\_534.sin\_family=AF\_INET;

sa\_addr\_534.sin\_addr.s\_addr=INADDR\_ANY;

sa\_addr\_534.sin\_port=htons(6000);

i\_534=bind(sockfd\_534,(struct sockaddr \*)&sa\_addr\_534,sizeof(sa\_addr\_534));

printf("test %d%d\n",sockfd\_534,i\_534);

listen(sockfd\_534,5);

length\_534=sizeof(cl\_addr\_534);

fd1\_534=accept(sockfd\_534, (struct sockaddr \*) &cl\_addr\_534,&length\_534);

recv(fd1\_534, &buf\_534, sizeof(buf\_534), 0);

while(buf\_534!=0)

{

int rem\_534=buf\_534%10;

rev\_534=rev\_534\*10+rem\_534;

buf\_534=buf\_534/10;

}

send(fd1\_534, &rev\_534, sizeof(rev\_534), 0);

close(fd1\_534);

}

**OUTPUT -:**

/\*WAP to create a TCP socket through which client will send an integer array to the server , server will sort the array and send back to the client . Client will display the sorted array . \*/

**CLIENT -:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

#include<unistd.h>

int main()

{

int i\_534,sockfd\_534,buf\_534[100],n\_534;

struct sockaddr\_in sa\_addr\_534;

sockfd\_534=socket(AF\_INET,SOCK\_STREAM,0);

sa\_addr\_534.sin\_family=AF\_INET;

sa\_addr\_534.sin\_addr.s\_addr=inet\_addr("127.0.0.1"); //Loop back IP address

sa\_addr\_534.sin\_port=htons(6000);

memset(sa\_addr\_534.sin\_zero, '\0', sizeof sa\_addr\_534.sin\_zero);

i\_534=connect(sockfd\_534,(struct sockaddr \*)&sa\_addr\_534,sizeof(sa\_addr\_534));

printf("RAj , Enter the size of the array\n");

scanf("%d",&n\_534);

printf("RAj , Enter the input into the array\n");

for(int i\_534=0; i\_534<n\_534; i\_534++)

{

scanf("%d",&buf\_534[i\_534]);

}

send(sockfd\_534, buf\_534, 100, 0);

send(sockfd\_534, &n\_534, sizeof(n\_534),0);

recv(sockfd\_534, buf\_534, 100, 0);

for(int i\_534=0; i\_534<n\_534; i\_534++)

{

printf("%d ",buf\_534[i\_534]);

}

printf("\n");

close(sockfd\_534);

}

**SERVER -:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

#include<unistd.h>

int main()

{

int sockfd\_534,fd1\_534, length\_534,i\_534,buf\_534[100],n\_534;

struct sockaddr\_in sa\_addr\_534,cl\_addr\_534;

sockfd\_534=socket(AF\_INET,SOCK\_STREAM,0);

sa\_addr\_534.sin\_family=AF\_INET;

sa\_addr\_534.sin\_addr.s\_addr=INADDR\_ANY;

sa\_addr\_534.sin\_port=htons(6000);

i\_534=bind(sockfd\_534,(struct sockaddr \*)&sa\_addr\_534,sizeof(sa\_addr\_534));

printf("test %d%d\n",sockfd\_534,i\_534);

listen(sockfd\_534,5);

length\_534=sizeof(cl\_addr\_534);

fd1\_534=accept(sockfd\_534, (struct sockaddr \*) &cl\_addr\_534,&length\_534);

for(i\_534=0; i\_534 < 100; i\_534++) buf\_534[i\_534] = 0;

recv(fd1\_534, buf\_534, 100, 0);

recv(fd1\_534, &n\_534 , sizeof(n\_534), 0);

for(int i\_534=0; i\_534<n\_534-1; i\_534++)

{

for(int j\_534=0; j\_534<n\_534-i\_534-1; j\_534++)

{

if(buf\_534[j\_534]>buf\_534[j\_534+1])

{

int temp\_534=buf\_534[j\_534];

buf\_534[j\_534]=buf\_534[j\_534+1];

buf\_534[j\_534+1]=temp\_534;

}

}

}

send(fd1\_534, buf\_534, 100, 0);

close(fd1\_534);

}

**OUTPUT -:**

/\*WAP to create a TCP socket through which two clients will connect to a server simultaneously. When the server will get an integer number from the first client , he will forward it to the second client and when it will get a number from the second client , he will forward it to the forst client. \*/

**CLIENT 1-:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

#include<unistd.h>

int main()

{

int i\_534,sockfd\_534;

int buf\_534;

struct sockaddr\_in sa\_addr\_534;

sockfd\_534=socket(AF\_INET,SOCK\_STREAM,0);

sa\_addr\_534.sin\_family=AF\_INET;

sa\_addr\_534.sin\_addr.s\_addr=inet\_addr("127.0.0.1");

sa\_addr\_534.sin\_port=htons(2995);

memset(sa\_addr\_534.sin\_zero, '\0', sizeof sa\_addr\_534.sin\_zero);

i\_534=connect(sockfd\_534,(struct sockaddr \*)&sa\_addr\_534,sizeof(sa\_addr\_534));

printf("Enter the number\n");

scanf("%d",&buf\_534);

send(sockfd\_534, &buf\_534, sizeof(buf\_534), 0);

recv(sockfd\_534, &buf\_534, sizeof(buf\_534), 0);

printf("%d\n", buf\_534);

close(sockfd\_534);

}

**CLIENT 2 -:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

#include<unistd.h>

int main()

{

int i\_534,sockfd\_534;

int buf\_534;

struct sockaddr\_in sa\_addr\_534;

sockfd\_534=socket(AF\_INET,SOCK\_STREAM,0);

sa\_addr\_534.sin\_family=AF\_INET;

sa\_addr\_534.sin\_addr.s\_addr=inet\_addr("127.0.0.1");

sa\_addr\_534.sin\_port=htons(3000);

memset(sa\_addr\_534.sin\_zero, '\0', sizeof sa\_addr\_534.sin\_zero);

i\_534=connect(sockfd\_534,(struct sockaddr \*)&sa\_addr\_534,sizeof(sa\_addr\_534));

printf("Enter the number\n");

scanf("%d",&buf\_534);

send(sockfd\_534, &buf\_534, sizeof(buf\_534), 0);

recv(sockfd\_534, &buf\_534, sizeof(buf\_534), 0);

printf("%d\n", buf\_534);

close(sockfd\_534);

}

**SERVER -:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<arpa/inet.h>

#include<fcntl.h>

#include<string.h>

#include<unistd.h>

int main()

{

int sockfd\_534,sockfd1\_534,fd1\_534,fd2\_534, length\_534,length1\_534,i\_534,j\_534;

int buf\_534,buf1\_534;

struct sockaddr\_in sa\_addr1\_534,cl\_addr1\_534;

struct sockaddr\_in sa\_addr\_534,cl\_addr\_534;

sockfd\_534=socket(AF\_INET,SOCK\_STREAM,0);

sockfd1\_534=socket(AF\_INET,SOCK\_STREAM,0);

sa\_addr\_534.sin\_family=AF\_INET;

sa\_addr1\_534.sin\_family=AF\_INET;

sa\_addr\_534.sin\_addr.s\_addr=INADDR\_ANY;

sa\_addr1\_534.sin\_addr.s\_addr=INADDR\_ANY;

sa\_addr\_534.sin\_port=htons(3000);

sa\_addr1\_534.sin\_port=htons(2995);

memset(sa\_addr\_534.sin\_zero, '\0', sizeof sa\_addr\_534.sin\_zero);

memset(sa\_addr1\_534.sin\_zero, '\0', sizeof sa\_addr1\_534.sin\_zero);

i\_534=bind(sockfd\_534,(struct sockaddr \*)&sa\_addr\_534,sizeof(sa\_addr\_534));

j\_534=bind(sockfd1\_534,(struct sockaddr \*)&sa\_addr1\_534,sizeof(sa\_addr1\_534));

printf("test %d%d\n",sockfd\_534,i\_534);

printf("test %d%d\n",sockfd1\_534,j\_534);

listen(sockfd\_534,5);

listen(sockfd1\_534,5);

length\_534=sizeof(cl\_addr\_534);

length\_534=sizeof(cl\_addr1\_534);

fd1\_534=accept(sockfd\_534, (struct sockaddr \*) &cl\_addr\_534,&length\_534);

fd2\_534=accept(sockfd1\_534, (struct sockaddr \*) &cl\_addr1\_534,&length1\_534);

recv(fd1\_534, &buf\_534, sizeof(buf\_534),0);

printf("%d",buf\_534);

recv(fd2\_534, &buf1\_534,sizeof(buf1\_534),0);

printf("%d",buf1\_534);

send(fd1\_534, &buf1\_534, sizeof(buf\_534),0);

send(fd2\_534, &buf\_534,sizeof(buf1\_534),0);

close(fd1\_534);

close(fd2\_534);

}