1. TCP SOCKETS

SERVER :

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

#define PORT 7575

int main() {
   int server_fd, new_socket;
   struct sockaddr_in address;
   int opt = 1;
   int addrlen = sizeof(address);
   char buffer[1024] = {0};
   const char *hello = "Hello from server";

   if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) == 0) {
       perror("socket failed");
       exit(EXIT_FAILURE);
   }

   if (setsockopt(server_fd, SOL_SOCKET, SO_REUSEADDR, &opt,
sizeof(opt))) {
       perror("setsockopt");
       exit(EXIT_FAILURE);
   }
   address.sin_family = AF_INET;
   address.sin_addr.s_addr = INADDR_ANY;
   address.sin_port = htons(PORT);

   if (bind(server_fd, (struct sockaddr *)&address, sizeof(address))
< 0) {
       perror("bind failed");
       exit(EXIT_FAILURE);
   }
   if (listen(server_fd, 3) < 0) {
       perror("listen");
       exit(EXIT_FAILURE);
   }
```

```
   if ((new_socket = accept(server_fd, (struct sockaddr *)&address,
(socklen_t*)&addrlen)) < 0) {
       perror("accept");
       exit(EXIT_FAILURE);
   }

   int valread = read(new_socket, buffer, 1024);
   printf("Message from client: %s\n", buffer);
   send(new_socket, hello, strlen(hello), 0);
   printf("Hello message sent\n");

   close(new_socket);
   close(server_fd);
   return 0;
}
```

OUTPUT:

```
rana:-/Desktop/lab$ gcc 1s.c

rana:-/Desktop/lab$./a.out

Message from client: Hello from client

Hello message sent
```

CLIENT :

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

#define PORT 7575

int main() {
   int sock = 0;
   struct sockaddr_in serv_addr;
   const char *hello = "Hello from client";
   char buffer[1024] = {0};

   if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
```

```
        printf("\n Socket creation error \n");
        return -1;
    }

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(PORT);

    if (inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr) <= 0) {
        printf("\nInvalid address/ Address not supported \n");
        return -1;
    }

    if (connect(sock, (struct sockaddr *)&serv_addr,
sizeof(serv_addr)) < 0) {
        printf("\nConnection Failed \n");
        return -1;
    }

    send(sock, hello, strlen(hello), 0);
    printf("Hello message sent\n");
    int valread = read(sock, buffer, 1024);
    printf("Message from server: %s\n", buffer);

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

OUTPUT:

```
rana:-/Desktop/lab$ gcc 1c.c

rana:-/Desktop/lab$./a.out

Hello message sent
Message from server: Hello from server
```

2. UDP SOCKET

SERVER :

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

#define PORT 8080

int main() {
   int sockfd;
   char buffer[1024];
   char *hello = "Hello from server";
   struct sockaddr_in servaddr, cliaddr;

   if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
       perror("socket creation failed");
       exit(EXIT_FAILURE);
   }

   memset(&servaddr, 0, sizeof(servaddr));
   memset(&cliaddr, 0, sizeof(cliaddr));

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

   if (bind(sockfd, (const struct sockaddr *)&servaddr,
sizeof(servaddr)) < 0) {
       perror("bind failed");
       close(sockfd);
       exit(EXIT_FAILURE);
   }

   int len, n;
   len = sizeof(cliaddr);

   while(1) {
       n = recvfrom(sockfd, buffer, 1024, MSG_WAITALL, (struct
sockaddr *)&cliaddr, &len);
       buffer[n] = '\0';
```

```
        printf("Client : %s\n", buffer);
        sendto(sockfd, hello, strlen(hello), MSG_CONFIRM, (const
struct sockaddr *)&cliaddr, len);
        printf("Hello message sent.\n");
        memset(buffer, 0, sizeof(buffer));
    }

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

OUTPUT:

```
rana:-/Desktop/lab$ gcc 2s.c

rana:-/Desktop/lab$./a.out

Client: Hello from client

Hello message sent.
```

CLIENT :

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

#define PORT 8080

int main() {
    int sockfd;
    char buffer[1024];
    char *hello = "Hello from client";
    struct sockaddr_in servaddr;

    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    memset(&servaddr, 0, sizeof(servaddr));

    servaddr.sin_family = AF_INET;
```

```
    servaddr.sin_port = htons(PORT);
    servaddr.sin_addr.s_addr = INADDR_ANY;

    int n, len;

    while(1) {
        sendto(sockfd, hello, strlen(hello), MSG_CONFIRM, (const
struct sockaddr *)&servaddr, sizeof(servaddr));
        printf("Hello message sent.\n");
        n = recvfrom(sockfd, buffer, 1024, MSG_WAITALL, (struct
sockaddr *)&servaddr, &len);
        buffer[n] = '\0';
        printf("Server : %s\n", buffer);
        memset(buffer, 0, sizeof(buffer));
        sleep(1);
    }

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

OUTPUT:

```
rana:-/Desktop/lab$ gcc 1s.c

rana:-/Desktop/lab$./a.out

Hello message sent.

Server : Hello from server
```

3. DECREMENT THE CHARACTER

SERVER :

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

#define PORT 4751

int main()
{
   int server_fd, new_socket;
   struct sockaddr_in address;
   int opt = 1;
   int addrlen = sizeof(address);
   char hello;

   if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) == 0)
   {
       perror("socket failed");
       exit(EXIT_FAILURE);
   }

   if (setsockopt(server_fd, SOL_SOCKET, SO_REUSEADDR, &opt,
sizeof(opt)))
   {
       perror("setsockopt");
       exit(EXIT_FAILURE);
   }
   address.sin_family = AF_INET;
   address.sin_addr.s_addr = INADDR_ANY;
   address.sin_port = htons(PORT);

   if (bind(server_fd, (struct sockaddr *)&address, sizeof(address))
< 0)
   {
       perror("bind failed");
       exit(EXIT_FAILURE);
   }
   if (listen(server_fd, 3) < 0)
   {
```

```
        perror("listen");
        exit(EXIT_FAILURE);
    }
    if ((new_socket = accept(server_fd, (struct sockaddr *)&address,
(socklen_t *)&addrlen)) < 0)
    {
        perror("accept");
        exit(EXIT_FAILURE);
    }

    int valread = read(new_socket, &hello, 1);
    printf("Message from client: %c\n", hello);
    close(new_socket);

    if ((new_socket = accept(server_fd, (struct sockaddr *)&address,
(socklen_t *)&addrlen)) < 0)
    {
        perror("accept");
        exit(EXIT_FAILURE);
    }

    hello = hello - 1;
    send(new_socket, &hello, strlen(&hello), 0);
    printf("Message sent\n");

    close(new_socket);
    close(server_fd);
    return 0;
}
```

OUTPUT:

```
rana:-/Desktop/lab$ gcc 3s.c

rana:-/Desktop/lab$./a.out

Message from client : d

Message sent
```

CLIENT 1 :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
```

```
#include <arpa/inet.h>

#define PORT 4751

int main() {
   int sock = 0;
   struct sockaddr_in serv_addr;
   char hello;

   if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
       printf("\n Socket creation error \n");
       return -1;
   }

   serv_addr.sin_family = AF_INET;
   serv_addr.sin_port = htons(PORT);

   if (inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr) <= 0) {
       printf("\nInvalid address/ Address not supported \n");
       return -1;
   }

   if (connect(sock, (struct sockaddr *)&serv_addr,
sizeof(serv_addr)) < 0) {
       printf("\nConnection Failed \n");
       return -1;
   }

   printf("Enter a character: ");
   scanf("%c", &hello);
   send(sock, &hello, strlen(&hello), 0);
   printf("Message sent\n");
   close(sock);
   return 0;
}
```

OUTPUT:

```
rana:-/Desktop/lab$ gcc 3c1.c

rana:-/Desktop/lab$./a.out

Enter a character : d

Message sent
```

CLIENT 2 :

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

#define PORT 4751

int main() {
   int sock = 0;
   struct sockaddr_in serv_addr;
   char hello;

   if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
       printf("\n Socket creation error \n");
       return -1;
   }
   serv_addr.sin_family = AF_INET;
   serv_addr.sin_port = htons(PORT);

   if (inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr) <= 0) {
       printf("\nInvalid address/ Address not supported \n");
       return -1;
   }

   if (connect(sock, (struct sockaddr *)&serv_addr,
sizeof(serv_addr)) < 0) {
       printf("\nConnection Failed \n");
       return -1;
   }
   int valread = read(sock, &hello, 1);
   printf("Message from server: %c\n", hello);
   close(sock);
   return 0;
}
```

OUTPUT:

```
rana:-/Desktop/lab$ gcc 3c2.c

rana:-/Desktop/lab$./a.out

Message from server : c
```