

PALINDROME CHECK USING TCP

SERVER

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

void main()
{
    struct sockaddr_in server, client;
    int s, n, sock;
    int flag, left, right;
    char buffer[20];

    s = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 2000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");

    bind(s, (struct sockaddr *)&server, sizeof server);
    listen(s, 1);
    n = sizeof client;
    sock = accept(s, (struct sockaddr *)&client, &n);

    for(;;)
    {
        recv(sock, buffer, sizeof buffer, 0);
        printf("\nThe string received is : %s");
        //Same code till here for palindrome, prime number and Armstrong

        if (strlen(buffer)==0) flag=1;
        else
        {
            flag = 1;
            left = 0;
            right = strlen(buffer)-1;
            while (left<right && flag)
            {
                if (buffer[left] != buffer[right])
                    flag = 0;
                else
                {

```

```

        left++;
        right++;
    }
}

send(sock, &flag, sizeof(int), 0);
break;
}
close(sock);
close(s);
}

```

CLIENT

```

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

void main()
{
    struct sockaddr_in server, client;
    int s, flag;
    char buffer[20];

    s = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 2000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");

    connect(s, (struct sockaddr *)&server, sizeof server);

    for(;;)
    {
        printf("Enter string to check palindrome : ");
        scanf("%s", &buffer);

        printf("\nClient : %s", buffer);
        send(s, buffer, sizeof buffer, 0);
        recv(s, &flag, sizeof(int), 0);
    }
}

```

```
    if (flag == 1)
    {
        printf("\nServer : The string is a palindrome");
        break;
    }
    else
    {
        printf("\nServer : The string is not a palindrome");
        break;
    }
}
close(s);
}
```

PRIME NUMBER CHECK USING TCP

SERVER

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

void main()
{
    struct sockaddr_in server, client;
    int s, n, sock;
    int flag = 1, i, num;
    char buffer[20];

    s = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 2000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");

    bind(s, (struct sockaddr *)&server, sizeof server);
    listen(s, 1);
    n = sizeof client;
    sock = accept(s, (struct sockaddr *)&client, &n);

    for (;;)
    {
        recv(sock, buffer, sizeof buffer, 0);
        //Same code till here for palindrome, prime number and Armstrong

        num = atoi(buffer);
        printf("\nThe number received is : %s", buffer);
        if (num == 1 || num == 0)
        {
            flag = 0;
        }
        else
        {
            for (i = 2; i <= num / 2; i++)
            {
                if (num % i == 0)
                {

```

```

        flag = 0;
        break;
    }
}
}
send(sock, &flag, sizeof(int), 0);
break;
}
close(sock);
close(s);
}

```

CLIENT

```

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

void main()
{
    struct sockaddr_in server, client;
    int s, flag;
    char buffer[20];

    s = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 2000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");

    connect(s, (struct sockaddr *)&server, sizeof server);

    for (;;)
    {
        printf("Enter number to check prime number : ");
        scanf("%s", &buffer);

        printf("\nClient : %s", buffer);
        send(s, buffer, sizeof buffer, 0);
        recv(s, &flag, sizeof(int), 0);
        if (flag == 1)
    }
}

```

```
{  
    printf("\nServer : The number is a prime number");  
    break;  
}  
else  
{  
    printf("\nServer : The number is not a prime number");  
    break;  
}  
}  
close(s);  
}
```

ARMSTRONG NUMBER CHECK USING TCP

SERVER

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

void main()
{
    struct sockaddr_in server, client;
    int s, n, sock;
    int flag = 1, i, num, len;
    char buffer[20];

    s = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 2000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");

    bind(s, (struct sockaddr *)&server, sizeof server);
    listen(s, 1);
    n = sizeof client;
    sock = accept(s, (struct sockaddr *)&client, &n);

    for (;;)
    {
        recv(sock, buffer, sizeof buffer, 0);
        //Same code till here for palindrome, prime number and Armstrong

        int remainder, sum = 0;
        num = atoi(buffer);
        int original_num = num;
        len = strlen(buffer);

        printf("\nThe number received is : %s", buffer);
        for (num; num>0; num/=10)
        {
            remainder = num % 10;
            sum += pow(remainder, len);
            num /= 10;
        }
    }
}
```

```

    }
    if (original_num == sum)
        flag = 1;
    else
        flag = 0;
    send(sock, &flag, sizeof(int), 0);
    break;
}
close(sock);
close(s);
}

```

CLIENT

```

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

void main()
{
    struct sockaddr_in server, client;
    int s, flag;
    char buffer[20];

    s = socket(AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 2000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");

    connect(s, (struct sockaddr *)&server, sizeof server);

    for (;;)
    {
        printf("Enter number to check armstrong number : ");
        scanf("%s", &buffer);

        printf("\nClient : %s", buffer);
        send(s, buffer, sizeof buffer, 0);
        recv(s, &flag, sizeof(int), 0);
        if (flag == 1)
        {
            printf("\nServer : The number is an armstrong number");

```



```
        break;
    }
    else
    {
        printf("\nServer : The number is not an armstrong number");
        break;
    }
}
close(s);
}
```