PALINDROME CHECK

SERVER

```c
#include <arpa/inet.h>
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/stat.h>
#include <sys/types.h>
void main()
{
    struct sockaddr_in server, client;
    int s, n, left, right, flag, len;
    char b1[10], b2[10];
    s = socket(AF_INET, SOCK_DGRAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 3000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");
    bind(s, (struct sockaddr *)&server, sizeof(server));
    n = sizeof(client);
    while (1)
    {
        recvfrom(s, b1, sizeof(b1), 0, (struct sockaddr *)&client, &n);

        len = strlen(b1);
        left = 0;
        right = len-1;
        flag = 1;

        while (left<=right)
        {
            if (b1[left]!=b1[right])
            {
                flag = 0;
                break;
            }
            else
            {
                left++;
                right--;
            }
        }

        printf("\nReceived string : %s", b1);
        sendto(s, &flag, sizeof(int), 0, (struct sockaddr *)&client, n);
    }
}
```

CLIENT

```c
#include <arpa/inet.h>
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/stat.h>
#include <sys/types.h>
void main()
{
    struct sockaddr_in server, client;
    int s, n, flag;
    char b2[10];
    s = socket(AF_INET, SOCK_DGRAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 3000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");
    n = sizeof(server);
    while (1)
    {
        printf("\nEnter word to check palindrome : ");
        scanf("%s", &b2);

        sendto(s, b2, sizeof(b2), 0, (struct sockaddr *)&server, n);
        recvfrom(s, &flag, sizeof(int), 0, NULL, NULL);

        if (flag == 1)
            printf("\nServer : It is a palindrome");
        else
            printf("\nServer : It is not a palindrome");
    }
}
```

PRIME NUMBER CHECK

SERVER

```c
#include <arpa/inet.h>
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <stdlib.h>
void main()
{
    struct sockaddr_in server, client;
    int s, n, i, num, flag;
    char b1[10], b2[10];
    s = socket(AF_INET, SOCK_DGRAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 3000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");
    bind(s, (struct sockaddr *)&server, sizeof(server));
    n = sizeof(client);
    while (1)
    {
        recvfrom(s, b1, sizeof(b1), 0, (struct sockaddr *)&client,&n);
        num = atoi(b1);
        flag = 1;

        if (num == 1 || num == 0)
            flag = 0;

        for (i=2; i<=num/2; i++)
        {
            if (num%i == 0)
            {
                flag = 0;
                break;
            }
        }

        printf("\nReceived number : %s", b1);
        sendto(s, &flag, sizeof(int), 0, (struct sockaddr *)&client, n);
    }
}
```

CLIENT

```c
#include <arpa/inet.h>
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/stat.h>
#include <sys/types.h>
void main()
{
    struct sockaddr_in server, client;
    int s, n, flag;
    char b2[10];
    s = socket(AF_INET, SOCK_DGRAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 3000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");
    n = sizeof(server);
    while (1)
    {
        printf("\nEnter number to check prime : ");
        scanf("%s", &b2);

        sendto(s, b2, sizeof(b2), 0, (struct sockaddr *)&server, n);
        recvfrom(s, &flag, sizeof(int), 0, NULL, NULL);

        if (flag == 1)
            printf("\nServer : It is a prime number");
        else
            printf("\nServer : It is not a prime number");
    }
}
```

ARMSTRONG NUMBER CHECK

SERVER

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

void main()
{
    struct sockaddr_in server, client;
    int s, n, i, num, flag, len, original, remainder, sum;
    char b1[10], b2[10];
    s = socket(AF_INET, SOCK_DGRAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 3000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");
    bind(s, (struct sockaddr *)&server, sizeof(server));
    n = sizeof(client);
    while (1)
    {
        recvfrom(s, b1, sizeof(b1), 0, (struct sockaddr *)&client, &n);
        len = strlen(b1);
        num = atoi(b1);
        original = num;
        sum = 0;

        while (num != 0)
        {
            remainder = num % 10;
            sum += pow(remainder,len);
            num /= 10;
        }
        if (sum == original)
            flag = 1;
        else
            flag = 0;

        printf("\nReceived number : %s", b1);
        sendto(s, &flag, sizeof(int), 0, (struct sockaddr *)&client, n);
    }
}
```

CLIENT

```c
#include <arpa/inet.h>
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/stat.h>
#include <sys/types.h>
void main()
{
    struct sockaddr_in server, client;
    int s, n, flag;
    char b2[10];
    s = socket(AF_INET, SOCK_DGRAM, 0);
    server.sin_family = AF_INET;
    server.sin_port = 3000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");
    n = sizeof(server);
    while (1)
    {
        printf("\nEnter number to check armstrong : ");
        scanf("%s", &b2);

        sendto(s, b2, sizeof(b2), 0, (struct sockaddr *)&server, n);
        recvfrom(s, &flag, sizeof(int), 0, NULL, NULL);

        if (flag == 1)
            printf("\nServer : It is an armstrong number");
        else
            printf("\nServer : It is not an armstrong number");
    }
}
```

WHEN COMPILING THE SERVER FILE :

gcc serverfilename.c -o s.out -lm