TITLE:

NAME:SAURABH MUKHERJEE

CLASS:BCSE 3

ROLL NO:001910501006

GROUP:A1

ASSIGNMENT NUMBER:4

SUBJECT: COMPUTER NETWORKING LAB

PROBLEM STATEMENT:

Implement CDMA with Walsh code.

DESCRIPTION:

Walsh Codes are most commonly used in the orthogonal codes of CDMA applications. These codes correspond to lines of a special square matrix called the Hadamard matrix. For a set of Walsh codes of length N, it consists of n lines to form a square matrix of $n \times n$ Walsh code.

Walsh matrix can be represented as W(2n)=[W(n) W(n) W(n) W(n')] where W(n) is the walsh matrix W(n).W(1) is assumed as [+1].After that one can recursively compute the values for any number of stations .

DESIGN:

The system has two major components :

1.**Sender** (The sender has n number of stations which have data with them to be sent)

2.**Receiver** (Server in our case)

We have used the socket programming in our case to create the client-server architecture .

There is an independent function call create_Walsh() which takes the number of stations as its input and then creates the Walsh encoding matrix .

The data is sent to the server which then decodes it.

The server also provides user choices which is meant for decoding the data of particular stations.

INPUT / OUTPUT FORMAT

Input

1.Client Side

As an input to the client side we have

a.Number of stations

b.Maximum length of the data each station can send

2.Server Side

a.Server takes the input of the station number for which the data is to be decoded

Output

1.Client Side

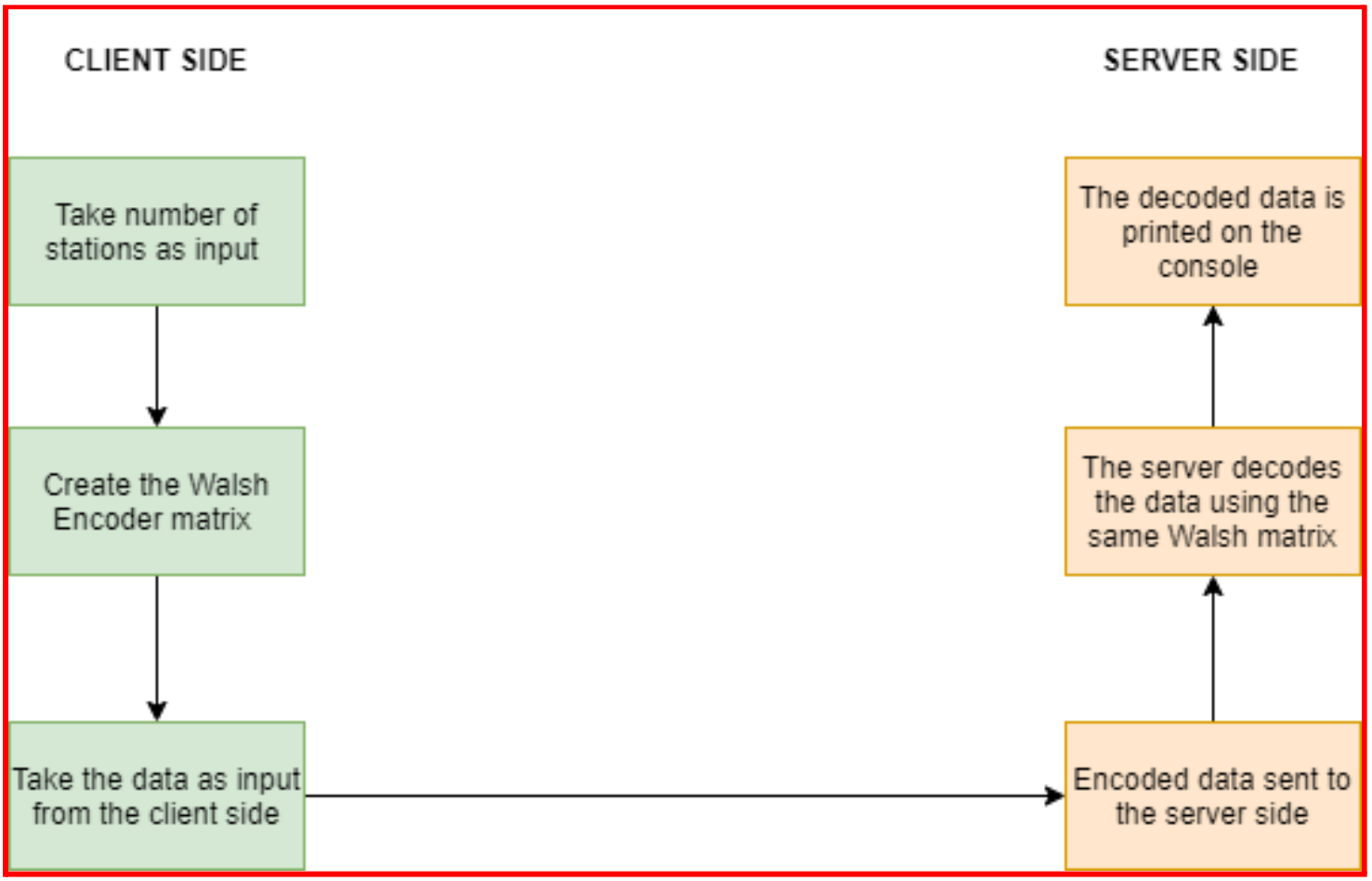
a.As a client side output we print the Walsh encoding matrix

b.We also print the final data after encoding using the CDMA technique

2.Server Side

a. The output on the server side consists of the decoded data which is decoded by using the same technique

FLOW OF DATA FROM THE CLIENT SIDE TO THE SERVER SIDE



IMPLEMENTATION

THE WALSH MATRIX METHOD:

```
def createWalsh(r):
     global walsh
     walsh=[[int(bin(x&y),13)%2 or -1 for x in range(r)]for y in range(r)]
```

THE BIT ENCODING METHOD:

```
def Convert(x):
    if x=="1":return 1
    elif x=="0":return -1
    else: return 0
```

THE ENCODING LOGIC:

```
max_length=int(input("Enter the maximum length of the data to be sent :"))
data=[input("Enter data for station:") for i in range(stations)]
for i in range(stations):
    diff=max_length-len(data[i])
    if diff>0:
        data[i]+=diff*('_')
    print("The new data for station ",i+1,":",data[i])

code=[walsh[i] for i in range(stations)]
print("Code for encoding the data by multiplying it with the data:")
for i in range(stations):
    code[i]=len(data[i])*code[i]
    print(code[i])

for i in range(stations):
    k=0
    for j in range(len(code[i])):
        if j%len(walsh[i])==0 and j!=0:k+=1
        if(data[i][k]=="0"):
```

```
            code[i][j]=-code[i][j]
        elif data[i][k]=="_":
            code[i][j]=0
    print("CODE :",code[i])
```

THE DECODING LOGIC:

```
def Decode(number_of_stations,data_recvd):
    r_value=2**(math.ceil(math.log(number_of_stations,2)))
    createWalsh(r_value)

    while True:
        Data=""
        while True:
            Station_Num=int(input("Enter the station number for which data
needs to be decoded:"))
            if(Station_Num>number_of_stations):
                print("--Please enter a valid station number--")
            else:
                break
        code=walsh[Station_Num-1]
        i=0
        while i<len(data_recvd):
            sum=0
            flag=False
            for k in range(len(code)):
                sum+=code[k]*data_recvd[i]
                i+=1
            if sum<0:flag=True
            sum=abs(sum)
            sum=sum//number_of_stations
            if sum==0:
                Data+="_"
            elif flag==True:
                Data+="0"
            else:Data+="1"
```

```
        print("The decoded data is :",Data)
        exit_code=int(input("Press 0 to exit"))
        if exit_code==0:
            exit()
```

—----------RESULTS —----------

For each test case we have taken the number of stations as an input from the user .The code generated for each of the test case is shown below along with the corresponding encoder matrix .The number of stations as well as the final data is sent to the server side as a list which is then abstracted and decoded using the same technique.

TEST CASE 1

CLIENT SIDE

```
...Waiting for connection response from the Server Side...
Enter the number of stations:3
1 1 1 1
1 -1 1 -1
1 1 -1 -1
1 -1 -1 1
Enter the maximum length of the data to be sent :5
Enter data for station:110_
Enter data for station:110
Enter data for station:0111
The new data for station  1 : 110__
The new data for station  2 : 110__
The new data for station  3 : 0111_
Code for encoding the data by multiplying it with the data:
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1]
[1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1]
CODE : [1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0]
CODE : [1, -1, 1, -1, 1, -1, 1, -1, -1, 1, -1, 1, 0, 0, 0, 0, 0, 0, 0, 0]
CODE : [-1, -1, 1, 1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 0, 0, 0, 0]
```

```
[3, 1, -1, 3, 1, 3, 1, 1, -1, -1, 1, -3, -1, 1, 1, -1, -1, 0, 0, 0, 0]
Transmission is successfull
SERVER SIDE :
Socket is listening..
Connected to: 127.0.0.1:63102
[1, -1, 3, 1, 3, 1, 1, -1, -1, 1, -3, -1, 1, 1, -1, -1, 0, 0, 0, 0]
Enter the station number for which data needs to be decoded:4
--Please enter a valid station number--
Enter the station number for which data needs to be decoded:2
The decoded data is : 110__
Press 0 to exit1
Enter the station number for which data needs to be decoded:1
The decoded data is : 110__
Press 0 to exit3
Enter the station number for which data needs to be decoded:3
The decoded data is : 0111_
Press 0 to exit0
```

TEST CASE 2

```
CLIENT SIDE
...Waiting for connection response from the Server Side...
Enter the number of stations:2
1 1
1 -1
Enter the maximum length of the data to be sent :5
Enter data for station:110
Enter data for station:1001
The new data for station  1 : 110__
The new data for station  2 : 1001_
Code for encoding the data by multiplying it with the data:
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
[1, -1, 1, -1, 1, -1, 1, -1, 1, -1]
CODE : [1, 1, 1, 1, -1, -1, 0, 0, 0, 0]
CODE : [1, -1, -1, 1, -1, 1, 1, -1, 0, 0]
[2, 2, 0, 0, 2, -2, 0, 1, -1, 0, 0]
```

```
Transmission is successfull

SERVER SIDE
Socket is listening..
Connected to: 127.0.0.1:57609
[2, 0, 0, 2, -2, 0, 1, -1, 0, 0]
Enter the station number for which data needs to be decoded:1
The decoded data is : 110__
Press 0 to exit1
Enter the station number for which data needs to be decoded:2
The decoded data is : 1001_
Press 0 to exit0
```

TEST CASE 3

```
CLIENT SIDE
...Waiting for connection response from the Server Side...
Enter the number of stations:5
1 1 1 1 1 1 1 1
1 -1 1 -1 1 -1 1 -1
1 1 -1 -1 1 1 -1 -1
1 -1 -1 1 1 -1 -1 1
1 1 1 1 -1 -1 -1 -1
1 -1 1 -1 -1 1 -1 1
1 1 -1 -1 -1 -1 1 1
1 -1 -1 1 -1 1 1 -1
Enter the maximum length of the data to be sent :8
Enter data for station:110
Enter data for station:1111
Enter data for station:0101_
Enter data for station:11001_
Enter data for station:11___
The new data for station  1 : 110_____
The new data for station  2 : 1111____
The new data for station  3 : 0101____
```

The new data for station 4 : 11001___

The new data for station 5 : 11______

Code for encoding the data by multiplying it with the data:

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

[1, -1, 1, -1]

[1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1]

[1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1]

[1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1]

CODE : [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

CODE : [1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

CODE : [-1, -1, 1, 1, -1, -1, 1, 1, 1, 1, -1, -1, 1, 1, -1, -1, -1, -1, 1, 1, -1, -1, 1, 1, 1, 1, -1, -1, 1, 1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

CODE : [1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, -1, 1, 1, -1, 1, -1, -1, 1, 1, -1, -1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

CODE : [1, 1, 1, 1, -1, -1, -1, -1, 1, 1, 1, 1, -1, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

[5, 3, -1, 3, 3, 1, -3, 1, 1, 5, 1, 1, 1, 3, -1, -1, -1, -2, -2, 2, -2, -2, -2, 2, -2, 1, 1, 1, -3, 1, 1, 1, -3, 1, -1, -1, 1, 1, -1, -1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Transmission is successfull

SERVER SIDE

```
Socket is listening..

Connected to: 127.0.0.1:55669

[3, -1, 3, 3, 1, -3, 1, 1, 5, 1, 1, 1, 3, -1, -1, -1, -2, -2, 2, -2, -2, -2,
2, -2, 1, 1, 1, -3, 1, 1, 1, -3, 1, -1, -1, 1, 1, -1, -1, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Enter the station number for which data needs to be decoded:3

The decoded data is : 0101____

Press 0 to exit1

Enter the station number for which data needs to be decoded:2

The decoded data is : 1111____

Press 0 to exit2

Enter the station number for which data needs to be decoded:7

--Please enter a valid station number--

Enter the station number for which data needs to be decoded:1

The decoded data is : 110_____

Press 0 to exit0
```

ANALYSIS:

The matrix for the Walsh table is created by using the logic of evil numbers and checking the parity of the bitwise AND of the row and column coordinates at any matrix with the 0 based indexing .If the parity is even we keep it -1 else if the parity is odd we keep our matrix value as 1.