```
clc;
clear all;
close all;
disp('Enter the data for the mu mimo single downlink system-----');
NumOfTransmitAntennas = input('\nEnter the values for number of transmit antennas Nt:');
NumOfReceiveAntennasPerUser = input('\nnumber of receive antennas per user Nr:');
VarianceSq = input('\nEnter the value for the variance square for the complex gaussian
zero mean random variables :');
NumOfUsers = input('\nEnter the value for total number of users :');
SNRindB = input('\nEnter the value for SNR in dB :');
Nt = NumOfTransmitAntennas;
Nr = NumOfReceiveAntennasPerUser;
v = VarianceSq;
k = NumOfUsers;
SNR = power(10,SNRindB/10);
Ebs =SNR * v;
rx = zeros(1,k*Nr);
user = zeros(1,k*Nr);
for i = 1:(k*Nr)
    rx(i) = i;
    user(i) =floor( (i-1)/Nr) + 1;
end
UserId = containers.Map(rx,user);
ChannelMatrix = sqrt(1/2)*randn(Nr,Nt,k) + sqrt(1/2)*randn(Nr,Nt,k)*1i;
S=[];
U=[];
l=zeros(1,Nt);
H_tilda=[];
Cmax=0;
Cr = zeros(1,length(rx));
flag=1;
phase=1;

while flag == 1
    for r = rx
        Stmp = union(S,r);
        W = [];
        H = H_tilda;
        ltmp = l;
        u = UserId(r);
        r_id = r - ((u-1)*Nr);
        Utmp = union(U,u);
        H(u) = [H(u);H(r_id,:,u)];   %index exceeds matrix dimension
        if phase == 1
            ltmp(u)= ltmp(u) + 1;
        end
        Ltmp(u) = sum(ltmp(u));
        for j = Utmp
            H1 = H(r_id,:,j)' * H(r_id,:,j);
            H2 = H_tilda(j)' * H_tilda(j) ;
            Wj(j) = eig(H1 , (Mj*v/Ej)*eye(Nt) + H2);
```

```
            for user = Utmp
                if(user ~= j)
                    H11 = H(r_id,:,user)' * H(r_id,:,user);
                    H22 = H_tilda(user)' * H_tilda(user) ;
                    Wj_t(j) = eig(H11 , (Mj*v/Ej)*eye(Nt) + H22);
                    Wj_tilda(j) = [Wj_tilda(user) Wj_t(user)];
                end
            end
            for l = 1:Ltmp(j)
               Hj = H(r_id,:,j);
               Dj = Wj' *  Hj' * Hj * Wj ;
               Num = Dj * Dj' ;
               Qj = Wj' * Hj' * Hj ;
               W = Wj_tilda * Wj_tilda' ;
               Dnum = ((Ltmp * v / Ebs) * Dj + ( Qj * W * Qj') );
               Numerator = Num(l,l);
               Denumerator = Dnum(l,l);
               SINR_j_l = Numerator / Denumerator ;
               C(j) = C(j) + log2( 1 + SINR_j_l ) ;
            end
        end
        Cr(r) = sum(C);
    end
    [r_max,r_bar] = max(Cr);
    if Cr(r_bar) > Cmax
        Cmax = Cr(r_bar);
        S = union(S,r_bar);
        u_bar = UserId(r_bar);
        R = setdiff(R,r_bar);
        U = union(U,u_bar);
        r_bar_ID = r_bar - ((U_bar-1)*Nr);
        H_tilda(u_bar) = [H_tilda(u_bar); H(r_bar_ID,:,U_bar) ];%error
        if phase == 1
            l(u_bar) = l(u_bar) + 1 ;
        end
    elseif phase == 1
        phase = 2;
    else
        flag = 0;
    end
end
disp('\nthe output of the SUBOPTIMAL ALGORITHM 1 are---- ');
disp('\nselected receive antennas are:',S);
disp('\nselected users are:',U);
disp('\ntotal data streams to be transmitted are:',l);
% ----------------------END OF PROGRAM-------------------------------------
```