```
function [SumCapacity,SelectedReceiveAntenna,SelectedUser,DataStreams ] = SuboptimalAlgorithm2Final( NumOfTransmitAntennas,NumOfReceiveAntennasPerUser, VarianceSq, NumOfUsers,SNRindB )

% Delcaration

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

% Mapping of user and receive antennas

UserId = containers.Map(rx,user);

% generating full channel matrix

ChannelMatrix = sqrt(1/2)*randn(Nr,Nt,k) + sqrt(1/2)*randn(Nr,Nt,k)*1i;

% Iitialization

R = rx;
S = [];
U = [];
L = 0;
H_tilda = zeros(Nr,Nt);
W = [];
Cmax = 0;
flag = 1;
Cr = zeros(1,Nr*k);
Wr = [];

% Algorithm Starts

while flag == 1
    Cr = zeros(1,Nr*k);
    if L < Nt
        for r = R
            Csum = 0;
            Stmp = union(S,r);
            Ltmp = L+1;
            H = H_tilda' * H_tilda ;
            u = UserId(r);
```

```
 50              r_id = r - ((u-1)*Nr);
 51              hr = ChannelMatrix(r_id,:,u);
 52
 53              % generating the Precoding matrix for receive antenna r
 54
 55              Wr(:,r) = eigs(inv(( Ltmp * v / Ebs )*eye( size(H,2) ) + H ) * ( hr'*
hr)  , Nt) ;
 56              Wtmp = [W Wr(:,r)];
 57              for i = Stmp
 58                  ui =UserId(i);
 59                  i_id = i-((ui-1)*Nr);
 60                  hi = ChannelMatrix(i_id,:,ui) ;
 61                  hiWtmpi = (norm(hi * Wr(:,i))) ^ 2;
 62                  hiWtmp = 0;
 63                  for l_bar = Stmp
 64                      if l_bar ~= i
 65
 66                          % generating the Tilda of Precoding matrix for user
 67                          % r
 68
 69                          hiWtmp = hiWtmp + (norm(hi * Wr(:,l_bar))) ^ 2;
 70                      end
 71                  end
 72
 73                  % calculating the Sum capacity for receive antenna r
 74
 75                  Csum = Csum + log2( 1 + ( (hiWtmpi) / ( ( Ltmp * v / Ebs ) + ( hiWtmp
) ) ) ) ) ;
 76              end
 77              Cr(r) = Csum ;
 78          end
 79
 80           % finding the receive antenna which provides maximum sum capacity
 81
 82          [r_val,r_bar] = max(Cr);
 83          if Cr(r_bar) > Cmax
 84              Cmax = Cr(r_bar);
 85              S = union(S,r_bar);
 86              U = union(U,UserId(r_bar));
 87              R = setdiff(R,r_bar);
 88              L = L+1;
 89              W = [W Wr(:,r_bar)];
 90              r_bar_id = r_bar - ((UserId(r_bar) -1)*Nr) ;
 91              Hr = ChannelMatrix(r_bar_id,:,UserId(r_bar));
 92              H_tilda( ~any(H_tilda,2), : ) = [];
 93
 94              % updating H tilda matrix
 95
 96              H_tilda = [H_tilda;Hr];
 97          else
 98              flag = 0;
```

```
        end
    else
        flag = 0;
    end
end

% generaing the required output

SumCapacity = Cmax;
SelectedReceiveAntenna = S;
SelectedUser = U;
DataStreams = L;

end
% ----------------------END OF PROGRAM-----------------------------------
```