Experiment 6

Q1.

```
clear
clc
r=input('Enter radius of conductor=');
n=input('Enter number of identical strands=');
GMR=r*(exp(-0.25)*2^6*3^(6/7)*2^(6/7))^(1/n);
fprintf('GMR = %5.4f \n', GMR)
```

Q2.

```
clear
clc
Ds = input('Geometric Mean Radius = ');
dia = input('Conductor diameter= '); r=dia/2;
nb=0;
while nb ~= 1 & nb ~= 2 & nb ~= 3 & nb ~= 4
nb = input('No. of bundled cond. (enter 1 for single cond.) = ');
d = input('Bundle spacing = ');
if nb ~= 1 & nb ~= 2 & nb ~= 3 & nb ~= 4
disp('You can only enter 1 or 2 or 3 or 4'), end
end
if nb == 1, Dsb = Ds; rb = r;
elseif nb == 2, Dsb = (d*Ds)^(1/2); rb = (d*r)^(1/2);
elseif nb == 3, Dsb = (d^2*Ds)^(1/3); rb = (d^2*r)^(1/3);
elseif nb == 4, Dsb = 2^0.125*(d^3*Ds)^(1/4); rb = 2^0.125*(d^3*r)^(1/4);
end
```

Q3.

```
function [GMD, GMRL, GMRC] = gmd
clc, clear
par = [
' Parameters of transposed transmission lines                '
'                                                            '
'     Number of three-phase circuits            Enter       '
'     ------------------------------            -----       '
'     Single-circuit line                         1         '
'     Double-circuit vertical configuration       2         '
'     Double-circuit horizontal configuration     3         '
'     To quit                                     0         '
'                                                            '];
disp(par)

nc = -1;
  while nc ~=1 & nc ~= 2 & nc ~=3 & nc~=0
  nc = input('Select number of menu ');
    if nc ~= 1 & nc ~= 2 & nc ~=3 & nc~=0
     disp('Enter 1, 2 3 or 0'), end

  end
fprintf(' \n')
clc
if nc == 0, return, end
if nc == 1
   ckt1 = [
   '              b                                             '
   '              O                          For spacing unit use '
   '             / \                         m within quotes or   '
   '            /   \                        ft within quotes.    '
   '           /     \                                           '
   '         D12     D23                                         '
   '         /         \            a           b          c     '
   '        /           \           O----D12----O----D23----O     '
```

```
   '      a/             \c                                    '
   '      O------D13------O    OR     ----------D13----------  '];
   disp(ckt1)

   unit ='z';
   while strcmp(unit,'m')~=1 & strcmp(unit,'M') ~=1 & strcmp(unit,'ft')~=1 &
strcmp(unit,'FT') ~=1
      unit = input('Enter spacing unit within quotes ''m'' or ''ft''-> ');
   if strcmp(unit,'m')~=1 & strcmp(unit,'M') ~=1 & strcmp(unit,'ft')~=1 &
strcmp(unit,'FT') ~=1
      disp('Incorrect spacing unit, try again'),end
   end

   space = 0;
   while length(space) ~= 3
   space = input('Enter row vector [D12, D23, D13] = ');
     if length(space) ~= 3
     disp(' Values of D12, D23, D13 must be entered within brackets, try again.'),
end
   end
   D12 = space(1); D23 = space(2);D13=space(3);
   GMD = (D12*D23*D13)^(1/3);
elseif nc == 2
   ckt2 = [
   '                          a             c`(a`)                    '
   ' Circuit Arrangements     O-----S11-----O         For spacing unit use '
   ' --------------------     |                       m within quotes or   '
   ' (1)  abc-c`b`a`        H12                       ft within quotes.    '
   ' (2)  abc-a`b`c`      b  |               b`(b`)                       '
   '                      O--------S22--------O                          '
   '                         |                                          '
   '                      H23                                           '
   '                      c|           a`(c`)                            '
   '                       O----S33----O                                '];
   disp(ckt2)
   nph = 0;
   while nph ~= 1 & nph ~= 2
   nph = input('Enter (1 or 2)  -> ');
   end
   fprintf('  \n')
   unit = 0;
   while strcmp(unit,'m')~=1 & strcmp(unit,'M') ~=1 & strcmp(unit,'ft')~=1 &
strcmp(unit,'FT') ~=1
     unit = input('Enter spacing unit within quotes ''m'' or ''ft''-> ');
   if strcmp(unit,'m')~=1 & strcmp(unit,'M') ~=1 & strcmp(unit,'ft')~=1 &
strcmp(unit,'FT') ~=1
     disp('Incorrect spacing unit, try again'),end
   end
   S  =  0;
   while length(S) ~= 3
   S = input('Enter row vector [S11, S22, S33] = ');
     if length(S) ~= 3
     disp(' Values of S11, S22, S33 must be entered within brackets, try again.'),
end
   end
   H =  0;
   while length(H) ~= 2
   H = input('Enter row vector [H12, H23] = ');
     if length(H) ~= 2
     disp('Values of H12, H23 must be entered within brackets, try again.'), end
   end
   S11 = S(1); S22 = S(2); S33 = S(3); H12 = H(1); H23 = H(2);
   a1 = -S11/2 + j*H12;
   b1 = -S22/2 + j*0;
   c1 = -S33/2 - j*H23;
   if nph == 1
     a2 = S33/2 - j*H23;
     b2 = S22/2 + j*0;
```

```
      c2 = S11/2 + j*H12;
      elseif nph == 2
      a2 = S11/2 + j*H12;
      b2 = S22/2 + j*0;
      c2 = S33/2 - j*H23;
   end

elseif nc == 3
   ckt3 = [
   '        a         b         c             a`(c`)    b`(b`)   c`(a`) '
   '        O---D12---O---D23---O-----S11------O---D12---O---D23---O    '
   '         --------D13--------                --------D13--------     '
   '                                                                     '
   ' Circuit Arrangements                       For spacing unit use   '
   ' --------------------                       m within quotes or     '
   ' (1) abc-a`b`c`                             ft within quotes.      '
   ' (2) abc-c`b`a`                                                    '];
   disp(ckt3)
   nph = 0;
   while nph ~= 1 & nph ~= 2
     nph = input('Enter (1 or 2)  -> ');
   end
   fprintf('  \n')
   unit = 0;
   while strcmp(unit,'m')~=1 & strcmp(unit,'M') ~=1 & strcmp(unit,'ft')~=1 &
strcmp(unit,'FT') ~=1
     unit = input('Enter spacing unit within quotes ''m'' or ''ft''-> ');
   if strcmp(unit,'m')~=1 & strcmp(unit,'M') ~=1 & strcmp(unit,'ft')~=1 &
strcmp(unit,'FT') ~=1
     disp('Incorrect spacing unit, try again'),end
   end
   S  =  0;
   while length(S) ~= 3
   S = input('Enter row vector [D12, D23, D13] = ');
      if length(S) ~= 3
      disp(' Values of D12, D23, D13 must be entered within brackets, try again.'),
end
   end
   S11 = input('Enter Distance between two circuits, S11 = ');
   D12 = S(1); D23 = S(2); D13 = S(3);
   a1 = -(D13+S11/2);
   b1 = -(D23+S11/2);
   c1 = -S11/2;
     if nph == 1
     a2 = S11/2;
     b2 = D12+S11/2;
     c2 = D13+S11/2;
     elseif nph == 2
     a2 = D13+S11/2;
     b2 = D12+S11/2;
     c2 = S11/2;
     end
end
     if nc==2 | nc == 3
       Da1b1 = abs(a1 - b1);   Da1b2 = abs(a1 - b2);
       Da1c1 = abs(a1 - c1);   Da1c2 = abs(a1 - c2);
       Db1c1 = abs(b1 - c1);   Db1c2 = abs(b1 - c2);

       Da2b1 = abs(a2 - b1);   Da2b2 = abs(a2 - b2);
       Da2c1 = abs(a2 - c1);   Da2c2 = abs(a2 - c2);
       Db2c1 = abs(b2 - c1);   Db2c2 = abs(b2 - c2);

       Da1a2 = abs(a1 - a2);
       Db1b2 = abs(b1 - b2);
       Dc1c2 = abs(c1 - c2);
       DAB=(Da1b1*Da1b2* Da2b1*Da2b2)^0.25;
       DBC=(Db1c1*Db1c2*Db2c1*Db2c2)^.25;
       DCA=(Da1c1*Da1c2*Da2c1*Da2c2)^.25;
```

```
       GMD=(DAB*DBC*DCA)^(1/3);
     end

  unitc = 0;
  while strcmp(unitc,'cm')~=1 & strcmp(unitc,'CM') ~=1 & strcmp(unitc,'in')~=1 &
strcmp(unitc,'IN') ~=1
  unitc = input('Cond. size, bundle spacing unit: Enter ''cm'' or ''in''-> ');
  if strcmp(unitc,'cm')~=1 & strcmp(unitc,'CM') ~=1 & strcmp(unitc,'in')~=1 &
strcmp(unitc,'IN') ~=1
     disp('Incorrect conductor unit, try again'),end
  end
if unitc == 'cm' | unitc == 'CM'
dia = input('Conductor diameter in cm = '); r=dia/2;
Ds  = input('Geometric Mean Radius in cm = ');
elseif unitc == 'in' | unitc == 'IN'
dia = input('Conductor diameter in inch = '); r=dia/2;
Ds  = input('Geometric Mean Radius in inch = ');
end
nb = 0;
   while nb ~= 1 & nb ~= 2 & nb ~= 3 & nb ~= 4
   nb =  input('No. of bundled cond. (enter 1 for single cond.) = ');
     if nb ~= 1 & nb ~= 2 & nb ~= 3 & nb ~= 4
     disp('You can only enter 1 or 2 or 3 or 4'), end
     end
if nb > 1
%  fprintf(' \n')
   if unitc == 'cm' | unitc == 'CM'
     d = input('Bundle spacing in cm = ');
     elseif unitc == 'in' | unitc == 'IN'
     d = input('Bundle spacing in inch = ');
     end
else d = 0; end
if unit == 'm'| unit =='M'
     if unitc == 'cm'| unitc == 'CM', Ds = Ds/100; r = r/100; d = d/100;
     elseif unitc == 'in'| unitc == 'IN', Ds = 2.54*Ds/100; r = 2.54*r/100; d =
2.54*d/100;
     end
elseif unit == 'ft' | unit == 'FT'
     if unitc == 'in'| unitc == 'IN', Ds = Ds/12; r = r/12; d = d/12;
     elseif unitc == 'cm'| unitc == 'CM', Ds = Ds/(2.54*12); r = r/(2.54*12); d =
d/(2.54*12);
     end
end
if nb == 1, Dsb = Ds;  rb = r;
elseif nb == 2, Dsb = (d*Ds)^(1/2); rb = (d*r)^(1/2);
elseif nb == 3, Dsb = (d^2*Ds)^(1/3); rb = (d^2*r)^(1/3);
elseif nb == 4, Dsb = 2^0.125*(d^3*Ds)^(1/4); rb = 2^0.125*(d^3*r)^(1/4);
end
if nc == 1
GMRL = Dsb; GMRC = rb;
elseif nc == 2 | nc == 3
DSA=sqrt(Dsb*Da1a2);      rA = sqrt(rb*Da1a2);
DSB=sqrt(Dsb*Db1b2);      rB = sqrt(rb*Db1b2);
DSC=sqrt(Dsb*Dc1c2);      rC = sqrt(rb*Dc1c2);
GMRL=(DSA*DSB*DSC)^(1/3); GMRC = (rA*rB*rC)^(1/3);
end
fprintf('  \n\n')
if unit == 'm' | unit =='M'
fprintf(' GMD  = %8.5f m\n',GMD)
fprintf(' GMRL = %8.5f m',GMRL),  fprintf('   GMRC = %8.5f m\n', GMRC)
elseif unit == 'ft' | unit == 'FT'
fprintf(' GMD  = %8.5f ft\n',GMD)
fprintf(' GMRL = %8.5f ft',GMRL), fprintf('   GMRC = %8.5f ft\n', GMRC)
end
```