

```

clear all
close all
clc

syms ks a

n = input('insert n(n is the number of columns/beams) >> ');
n1 = 2*n; % Dimension of the Matrix is 2 times the number of beams

X = sym(zeros(n1,n1)); % (sym) makes the matrix X compatible to hold
parameters such as ks and a in it

alpha = input('insert alpha ((n-1)th beam is (alpha) times (n)th beam) >> ');
f = zeros(1,n);

j = 1;
for i = n-1:-1:0
    f(j) = alpha^i;
    j = j + 1;
end
disp(f)
B = perms(f); %modelling permutation
disp(B)

rows = size(B,1); %rows is the number of permuted states
cols = size(B,2);

RmMAX = zeros(1,rows);

%%forming even columns (permutation doesn't affect EVEN columns)
for j = 2:2:n1
    if j<n1
        X(j,j-1) = 1;
        X(j,j) = -ks;
        X(j,j+1) = -1;
    elseif j == n1
        X(j,j-1) = 1;
        X(j,j) = -ks;
    end
end

%%forming odd columns (with regards to permutation)
for q = 1:rows
    for jj = 1:2:2*n-1
        if jj ==1
            X(jj,jj) = 1;
            X(jj,jj+1) = -B(q, ((jj+1)/2))*a*ks;
        else
            X(jj,jj-1) = B(q, (jj+1)/2)*a*ks;
            X(jj,jj) = 1;
            X(jj,jj+1) = -B(q, (jj+1)/2)*a*ks;
        end
    end
end
disp(X)
g=det(X);

```

```

f=subs(g,1);
Rm0=max(double(vpasolve(f,a)));
display(g)
display(f)
display(Rm)
RmMAX(q) = Rm0;
end
disp(RmMAX)
Rm = min(RmMAX);
disp(['Rm(min) = ' num2str(Rm)])

E = 29500;      %%unit=ksi
Abr1 = zeros(1,n);
Abr2 = zeros(1,n); %Abr2 is (beta) times Abr1

beta = input('insert beta (Abr2 is (beta) times Abr1)>> ');

j = 1;
for i = n-1:-1:0
    Ic=307;      %%Inertia moment  unit=in^4
    Lc=264;      %%Column Length  unit=in
    Lb=100;      %%Brace Length  unit=in
    Pcr = (pi^2*E*Ic)/(Lc)^2;
    Ks = Pcr/Lc;  %%unit=Kips/in
    Kn1 = Rm*Ks*alpha^i;  %%unit=Kips/in
    Kn2 = beta*Kn1;  %%unit=Kips/in
    AbrA=(Kn1*Lb)/E;  %%Ideal Brace Area  unit=in^2
    AbrB=(Kn2*Lb)/E;  %%Ideal Brace Area  unit=in^2
    Abr1(j) = AbrA;
    Abr2(j) = AbrB;
    j = j + 1;
end
display(Abr1)
display(Abr2)

```