

```

//Programme d'étude du flambage des poutres: Bertin & Rivoire
# include <iostream>
# include <math.h>
# define n 100

using namespace std;

void donnees(int p, float I[n], float a, float d, float b, float c,
             float f, float g, float h)
{
    float S[n];
    for (int i=0; i<=p; i++)
    { S[i]=(c+(d-i*d/p)*(a-c)/d)*b; }
    for (int i=0; i<=p+1; i++)
    { I[i]=S[i]*(((S[i]/b)*(S[i]/b)+(b*b))/3); }
    cout<<" Aire           Moment quadratique";
    cout<<endl;
    for (int i=0; i<=p; i++)
    { cout<<"["; cout<<S[i]; cout<<"]"; cout<<"["; cout<<I[i]; cout<<"]"; cout<<endl; }
    cout<<endl;
    cout<<endl;
}

void raideur(int p, float d, float KG[n][n], float K[n][n], float I[n], float O,
             float TK[n][n], float TKG[n][n], float P, float T[n][n])
{
    float E[n]; float M[n];
    for (int i=0; i<p; i++)
    {
        K[0][0]=(48*(i*d/p)*(i*d/p)*(i*d/p))/((d/p)*(d/p)*(d/p)*(d/p)*(d/p)*(d/p));
        K[0][1]=(24*(i*d/p)*(i*d/p)*(i*d/p))/((d/p)*(d/p)*(d/p)*(d/p)*(d/p))
        -(6*(i*d/p)*(i*d/p))/((d/p)*(d/p)*(d/p)*(d/p));
        K[1][0]=K[0][1];
        K[0][2]=-K[0][0];
        K[2][0]=K[0][2];
        K[0][3]=(24*(i*d/p)*(i*d/p)*(i*d/p))/((d/p)*(d/p)*(d/p)*(d/p)*(d/p))+
        (6*(i*d/p)*(i*d/p))/((d/p)*(d/p)*(d/p)*(d/p));
        K[3][0]=K[0][3];
        K[1][1]=(((6*(i*d/p))/((d/p)*(d/p))-1/(d/p))*((6*(i*d/p))/((d/p)*(d/p))
        -1/(d/p))*((6*(i*d/p))/((d/p)*(d/p))-1/(d/p))*((d/p)*(d/p)))/18;
        K[1][2]=(-24*(i*d/p)*(i*d/p)*(i*d/p))/((d/p)*(d/p)*(d/p)*(d/p)*(d/p))+
        (6*(i*d/p)*(i*d/p))/((d/p)*(d/p)*(d/p)*(d/p));
        K[2][1]=K[1][2];
        K[1][3]=(12*(i*d/p)*(i*d/p)*(i*d/p))/((d/p)*(d/p)*(d/p)*(d/p))
        -(i*d/p)/((d/p)*(d/p));
        K[3][1]=K[1][3];
        K[2][2]=K[0][0];
        K[2][3]=-((24*(i*d/p)*(i*d/p)*(i*d/p))/((d/p)*(d/p)*(d/p)*(d/p)*(d/p))+
        (6*(i*d/p)*(i*d/p))/((d/p)*(d/p)*(d/p)*(d/p)));
        K[3][2]=K[2][3];
        K[3][3]=(((6*(i*d/p))/((d/p)*(d/p))-1/(d/p))*((6*(i*d/p))/((d/p)*(d/p))
        +1/(d/p))*((6*(i*d/p))/((d/p)*(d/p))-1/(d/p))*((d/p)*(d/p)))/18;

        KG[0][0]=(36*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/(5*(d/p)*(d/p)*
        (d/p)*(d/p)*(d/p)*(d/p)))-(6*(i*d/p)*(i*d/p)*(i*d/p)/((d/p)*(d/p)*(d/p)
        *(d/p)))+(9*(i*d/p)/(4*(d/p)*(d/p)));
        KG[0][1]=(18*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/(5*(d/p)*(d/p)*
        (d/p)*(d/p)*(d/p)))-(3*(i*d/p)*(i*d/p)*(i*d/p)/(2*(d/p)*(d/p)
        *(d/p)*(d/p)))-(2*(i*d/p)*(i*d/p)/(d/p)*(d/p)*(d/p))+
        (3*(i*d/p)*(i*d/p)/(4*(d/p)*(d/p)))+(3*(i*d/p)/(8*(d/p)));
    }
}

```

```

KG[1][0]=KG[0][1];
KG[0][2]=-((36*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/(5*(d/p)*
(d/p)*(d/p)*(d/p)*(d/p)*(d/p)))+(6*(i*d/p)*(i*d/p)*(i*d/p)/((d/p)*
(d/p)*(d/p)*(d/p)))-(9*(i*d/p)/(4*(d/p)*(d/p))));
KG[2][0]=KG[0][2];
KG[0][3]=((18*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/(5*(d/p)*(d/p)*
(d/p)*(d/p)*(d/p)))+(3*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/(2*(d/p)*(d/p)*
(d/p)*(d/p)))-(2*(i*d/p)*(i*d/p)*(i*d/p)/((d/p)*(d/p)*(d/p)))
-(3*(i*d/p)*(i*d/p)/(4*(d/p)*(d/p)))+(3*(i*d/p)/(8*(d/p))));
KG[3][0]=KG[0][3];
KG[1][1]=((i*d/p)/16)+(9*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/
(5*(d/p)*(d/p)*(d/p)*(d/p)))-(3*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/
(2*(d/p)*(d/p)*(d/p)))-((i*d/p)*(i*d/p)*(i*d/p)/(6*(d/p)*(d/p)))+
((i*d/p)*(i*d/p)/(4*(d/p)));
KG[1][2]=-((18*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/(5*(d/p)*
(d/p)*(d/p)*(d/p)*(d/p)))+(3*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/
(2*(d/p)*(d/p)*(d/p)*(d/p)))+(2*(i*d/p)*(i*d/p)*(i*d/p)/((d/p)
*(d/p)*(d/p)))-(3*(i*d/p)*(i*d/p)/(4*(d/p)*(d/p)))-(3*(i*d/p)/(8*(d/p))));
KG[2][1]=K[1][2];
KG[1][3]=((i*d/p)/16)+(9*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/
(5*(d/p)*(d/p)*(d/p)*(d/p)))-(5*(i*d/p)*(i*d/p)*(i*d/p)/(6*(d/p)*(d/p)));
KG[3][1]=KG[1][3];
KG[2][2]=((36*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/(5*(d/p)*(d/p)*
(d/p)*(d/p)*(d/p)*(d/p)))-(6*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/((d/p)*(d/p)*
(d/p)*(d/p)))+(9*(i*d/p)/(4*(d/p)*(d/p))));
KG[2][3]=-((18*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/(5*(d/p)*(d/p)
*(d/p)*(d/p)*(d/p)))-(3*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/(2*(d/p)*(d/p)
*(d/p)*(d/p)))+(2*(i*d/p)*(i*d/p)*(i*d/p)/((d/p)*(d/p)*(d/p)))
+(3*(i*d/p)*(i*d/p)/(4*(d/p)*(d/p)))-(3*(i*d/p)/(8*(d/p))));
KG[3][2]=K[2][3];
KG[3][3]=((i*d/p)/16)+(9*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/
(5*(d/p)*(d/p)*(d/p)*(d/p)))+(3*(i*d/p)*(i*d/p)*(i*d/p)*(i*d/p)/
(2*(d/p)*(d/p)*(d/p)))-((i*d/p)*(i*d/p)*(i*d/p)/(6*(d/p)*(d/p)))
-((i*d/p)*(i*d/p)/(4*(d/p)));

//fabrication de la matrice K sur toute la longueur de la poutre.
for(int k=0; k<=3; k++)
{
    for(int j=0; j<=3; j++)
    {
        TK[2*i+k][2*i+j]+=K[k][j];
    }
}

for(int k=0; k<=3; k++)
{
    for(int j=0; j<=3; j++)
    {
        TKG[2*i+k][2*i+j]+=KG[k][j];
    }
}
cout<<"K = ";cout<<endl;
for(int o=0; o<2*p+2; o++)
{
    cout<<"  ";
    for(int s=0; s<2*p+2; s++)
    {
        cout<<"  "<<TK[o][s]<<"  ";
    }
    cout<<" ]"<<endl;
}

```

```

    }
    cout<<"KG = ";cout<<endl;
    for(int o=0; o<2*p+2; o++)
    {
        cout<<"  [";
        for(int s=0; s<2*p+2; s++)
        {
            cout<<"    "<<TKG[o][s]<<"  ";
        }
        cout<<" ]"<<endl;
    }
}

```

```

void EL(float A[n][n],int p,float B[n][n])
{
    for(int i=2;i<=2*p+1;i++)
    { for(int j=2;j<=2*p+1;j++)
      { B[i][j]=A[i][j];
      }
    }

    for(int i=2;i<=2*p+1;i++)
    { for(int j=2;j<=2*p+1;j++)
      { B[i-2][j-2]=B[i][j];
      }
    }
}

```

```

void RRM(float A[n][n],int p,float B[n][n])
{
    for(int i=1;i<=2*p;i++)
    { for(int j=1;j<=2*p;j++)
      { B[i][j]=A[i][j];
      }
    }
    for(int i=1;i<=2*p+1;i++)
    {B[i][2*p]=A[i][2*p+1];
    }
    for(int j=1;j<=2*p+1;j++)
    {B[2*p][j]=A[2*p+1][j];
    }
    for(int i=1;i<=2*p+1;i++)
    { for(int j=1;j<=2*p+1;j++)
      { B[i-1][j-1]=B[i][j];
      }
    }
    B[2*p-1][2*p-1]=A[2*p+1][2*p+1];
}

```

```

void ERM(float A[n][n],int p,float B[n][n])
{
    for(int i=2;i<=2*p;i++)
    { for(int j=2;j<=2*p;j++)
      { B[i][j]=A[i][j];
      }
    }
    for(int i=2;i<=2*p+1;i++)

```

```

        {B[i][2*p]=A[i][2*p+1];

    }
    for(int j=2;j<=2*p+1;j++)
    {B[2*p][j]=A[2*p+1][j];

    }
    for(int i=2;i<=2*p+1;i++)
    { for(int j=2;j<=2*p+1;j++)
      { B[i-2][j-2]=B[i][j];
        }
      }

    B[2*p-2][2*p-2]=A[2*p+1][2*p+1];

}

void EEM(float A[n][n],int p,float B[n][n])
{
    for(int i=2;i<=2*p;i++)
    { for(int j=2;j<=2*p;j++)
      { B[i][j]=A[i][j];
        }
      }

    for(int i=2;i<=2*p;i++)
    { for(int j=2;j<=2*p;j++)
      { B[i-2][j-2]=B[i][j];
        }
      }
}

void affichermat(float A[n][n], int N1, int N2, float u)
{
    for(int i=0; i<2*N1+u; i++)
    {
        cout<<"[";
        for(int j=0 ;j<2*N2+u; j++)
        {
            cout<<" " <<A[i][j]<<" ";
        }
        cout<<" ]"<<endl;
    }
}

void cholesky(float M[n][n], int p, float L[n][n],int u)
{ float s; float z;
  for(int i=0;i<2*p+u;i++)
  { for(int j=i;j>=0;j=j-1)
    {
        if(i==j)
        {
            if((i-1)<1){L[i][j]=M[i][j];}
            if((i-1)==1){s=L[i][1]*L[i][1];L[i][j]=pow((M[i][j]-s),
0.5);}
            if((i-1)>1){for(int k=1;k<=i-1;k++)
                        {
                            s+=L[i][k]*L[i][k];
                        }
                        L[i][j]=pow(M[i][j]-s,0.5);
                    }
        }
    }
  }
}

```

```

        if(i>j)
        {
            if((i-1)<1){z=0;L[i][j]=(M[i][j]-z)/L[i][i];}
            if((i-1)==1){z=L[i][1]*L[j][1];L[i][j]=(M[i][j]-z)/
            L[i][i];}
            if((i-1)>1){for(int k=1;k<=i-1;k++)
            {
                z+=L[i][k]*L[j][k];
            }
            L[i][j]=(M[i][j]-z)/L[i][i];
        }
    }
}

void transpmat(float C[n][n],float B[n][n],int p, float u)
{
    for (int i=0; i<2*p+u; i++)
    {for (int j=0; j<2*p+u; j++)
    {B[i][j]=C[j][i];}
    }
}

void produitmatrice(float A[n][n], float B[n][n],float C[n][n], int p, float u)
{
    for (int i = 0; i < 2*p+u; i++)
    {
        for (int j = 0; j < 2*p+u; j++)
        {
            for (int z = 0; z < 2*p+u; z++)
            {
                C[i][j] += A[i][z] * B[z][j];
            }
        }
    }
}

void InvdiagMat(int p, float Mat[n][n],float u)
{
    float temp[n][n];int i;
    for(i=0;i<2*p+u;i++)
    {
        for(int j=0;j<2*p+u;j++)
        {
            temp[i][j]=0;
        }
    }

    for(i=0;i<2*p+u;i++)
    {
        for(int j=0;j<2*p+u;j++)
        {
            temp[i][i]=1/Mat[i][i];
            if(j!=i)
            {
                temp[i][j]=-Mat[i][j]/Mat[i][i];
            }
        }
    }
}

```

```

    for(int k=0;k<2*p+u;k++)
    {
        if(k!=i)
        {
            temp[k][i]=Mat[k][i]/Mat[i][i];
        }
        if(j!=i &&k!=i)
        {
            temp[k][j]=Mat[k][j]-Mat[i][j]*Mat[k][i]/Mat[i][i];
        }
    }

    }

    for(int i=0;i<2*p+u;i++)
    {
        for(int j=0;j<2*p+u;j++)
        {
            Mat[i][j]=temp[i][j];
        }
    }
}

void prodmatvect( float A[n][n], float U[n], int a,float V[n] )
{
    int i;
    int j;

    for (i=0; i<a; i++)
    {V[i]=0;
        for (j=0; j<a; j++) V[i]+=A[i][j]*U[j];
    }
}

void valeupropre(float w[n][n], int p, int u, float e,float res,int d)
{
    float g[n]; float g1[n];float g2[n]; float P2; float P1;int H;H=0; float x;
    x=1000;
    float S; float Su;float LA[n];LA[n]=0;
    for(int f=1;f<2*p+u;f++)
    {g[0]=1;
        g[f]=0;
    }
    while(x>e)
    {P1=0; P2=0;
        prodmatvect(w,g,2*p+u,g1);

        cout<<"g1[0] = ";cout<<g1[0];cout<<endl;
        cout<<"g1[2*p+u-1] = ";cout<<g1[2*p+u-1];cout<<endl;

        for(int i=0; i<2*p+u; i++)
        {S += g1[i]*g1[i];
        }

        cout<<"S = ";cout<<S;cout<<endl;

        Su=pow(S,0.5);

        cout<<"Su = ";cout<<Su;cout<<endl;
    }
}

```

```

for(int o=0;o<2*p+u; o++)
    {g1[o] = g1[o]/Su;}

cout<<"g1[0] = ";cout<<g1[0];cout<<endl;
cout<<"g1[2*p+u-3] = ";cout<<g1[2*p+u-3];cout<<endl;
cout<<"g1[2*p+u-2] = ";cout<<g1[2*p+u-2];cout<<endl;
cout<<"g1[2*p+u-1] = ";cout<<g1[2*p+u-1];cout<<endl;

prodmatrix(w,g1,2*p+u,g2);

cout<<"g2[0] = ";cout<<g2[0];cout<<endl;
cout<<"g2[2*p+u-3] = ";cout<<g2[2*p+u-3];cout<<endl;
cout<<"g2[2*p+u-2] = ";cout<<g2[2*p+u-2];cout<<endl;
cout<<"g2[2*p+u-1] = ";cout<<g2[2*p+u-1];cout<<endl;

for(int t= 0; t<2*p+u;t++)
    {
        P1=g1[t]*g2[t]+P1;
    }
for(int t=0;t<2*p+u;t++)
    {
        P2=g1[t]*g1[t]+P2;
    }

    cout<<"P1 = ";cout<<P1;cout<<endl;
    cout<<"P2 = ";cout<<P2;cout<<endl;
    LA[H+1]=P1/P2;

x=pow((pow(LA[H+1]*LA[H+1],0.5) - pow(LA[H]*LA[H],0.5))*(pow(LA[H+1]*LA[H+1]
,0.5) - pow(LA[H]*LA[H],0.5)),0.5);
cout<<"x =";cout<<x;cout<<endl;

for(int i=0;i<2*p+u;i++)
    {g[i]=g1[i];}
res=LA[H+1];
H++;
d=H;
}
cout<<endl;cout<<"Le nombre d'iterations effectuées est de : ";
    cout<<endl;cout<<d;cout<<endl;
    cout<<endl;cout<<"La solution du problème est :";cout<<endl;
    cout<<res;cout<<endl;
}

int main()

{
    int J=0; while(J<1)
    {
        int p;float E;float I[n];float d;float KG[n][n];float a;float b;float c;
        float f;float g; float h;float K[n][n];float P;float TK[n][n];
        float TKG[n][n];
        float B[n][n];int w;float T[n][n];float C[n][n];float u;float L[n][n];
        float inVK[n][n];
        float Id[n][n];float LT[n][n];float IL[n][n]; float ILT[n][n];float A[n][n]
        ; float e;
        float LA[n];float res;int za;
        cout<<"-----";cout<<endl;
    }
}

```

```

cout<<"                programme par :";
cout<<endl;cout<<endl;
cout<<"                Bertin Morgan";
cout<<endl;
cout<<"                et";
cout<<endl;
cout<<"                Rivoire Pascal";
cout<<endl;cout<<endl;
cout<<endl;cout<<"vous allez rentrer les donnees du probleme";
cout<<endl;cout<<endl;cout<<"Module d'Young : (MPA) ";cin>>E;
cout<<endl;cout<<endl;cout<<"quel est le pas de discretisation?";
cout<<endl;cin>>p;
cout<<endl;cout<<"valeurs pour la base inferieure";
cout<<endl;cout<<"longueur?";cout<<endl;cin>>a;
cout<<endl;cout<<"largeur?";cout<<endl;cin>>b;
cout<<endl;cout<<"valeurs pour la base superieure (largeur = constante)";
cout<<endl;cout<<"longueur?";cout<<endl;cin>>c;
cout<<endl;cout<<"hauteur?";cout<<endl;cin>>d;cout<<endl;
cout<<endl;cout<<"precision des resultats, epsilon";cout<<endl;cin>>e;
cout<<endl;
cout<<endl;cout<<"condition limite";cout<<endl;
cout<<"cas 1 encastrement - libre : tapez 1 ";cout<<endl;
cout<<"cas 2 rotule - rotule mobile : tapez 2 ";cout<<endl;
cout<<"cas 3 encastrement - rotule mobile : tapez 3 ";cout<<endl;
cout<<"cas 4 encastrement - encastrement mobile : tapez 4 ";cout<<endl;
cin>>w;
donnees(p,I,a,d,b,c,f,g,h);system("pause");
raideur(p,d,KG,K,I,E,TK,TKG,P,T);
if(w==1){EL(TK,p,B);u=0;}
if(w==2){RRM(TK,p,B);u=0;}
if(w==3){ERM(TK,p,B);u=-1;}
if(w==4){EEM(TK,p,B);u=-1;}
if(w==1){EL(TKG,p,C);u=0;}
if(w==2){RRM(TKG,p,C);u=0;}
if(w==3){ERM(TKG,p,C);u=-1;}
if(w==4){EEM(TKG,p,C);u=-1;}
cout<<endl;cout<<" K(limite) = ";cout<<endl;
affichermat(B,p,p,u);system("pause");
cout<<endl;cout<<" KG(limite) = ";cout<<endl;
affichermat(C,p,p,u);system("pause");
cholesky(C,p,L,u); for (int i=0; i<2*p+u; i++)
    {for (int j=0; j<2*p+u; j++)
        {IL[i][j]=L[i][j];}
    }
cout<<endl;cout<<" L = ";cout<<endl;
affichermat(L,p,p,u); system("pause");
transpmat(L,LT,p,u);
InvdiagMat(p,L,u);
cout<<endl;cout<<" inverse de L = ";cout<<endl;
affichermat(L,p,p,u);system("pause");
produitmatrice(L,IL,ILT,p,u);
cout<<endl;cout<<" inverse de L*L = ";cout<<endl;
affichermat(ILT,p,p,u);system("pause");
InvdiagMat(p,LT,u);
cout<<endl;cout<<" inverse de LT = ";cout<<endl;
affichermat(LT,p,p,u);system("pause");
cout<<endl;cout<<" inverse de KG = ";cout<<endl;
produitmatrice(LT,L,inVK,p,u);
affichermat(inVK,p,p,u);system("pause");
produitmatrice(inVK,B,A,p,u);
cout<<endl;cout<<" INKG*K = ";cout<<endl;
affichermat(A,p,p,u);system("pause");

```



```
    valeupropri (A,p,u,e,res,za);

    system("pause");
    // cout<<"voulez vous recommencer?";cout<<endl;cout<<"Oui tapez 0 , non tapez 1"
    cout<<endl;cin>>J;cout<<endl;}

}
```