The original matrix `x[k][k]` is overwritten with its inverse matrix.

Therefore, `x[k][k]` is output as the global variable.

The computational procedure is as follows:

$A = \begin{pmatrix} 1 & 1 \\ 1 & 5 \end{pmatrix}$ is taken as an example.

A and I are put together.

$$(A \quad I) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{pmatrix}$$

The element of 2th line and 1st column becomes zero by subtracting the 1st line from the 2nd line.

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 5 & 0 & 1 \end{pmatrix} \quad \Longrightarrow \quad \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 4 & -1 & 1 \end{pmatrix}$$

Moreover, the element of 1st line and 2th column becomes zero by subtracting the 2nd line divided by 4 from the 1st line.

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 4 & -1 & 1 \end{pmatrix} \implies \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & -\frac{1}{4} & \frac{1}{4} \end{pmatrix} \implies \begin{pmatrix} 1 & 0 & \frac{5}{4} & -\frac{1}{4} \\ 0 & 1 & -\frac{1}{4} & \frac{1}{4} \end{pmatrix} = (I \ \ A^{-1})$$

Thus, the inverse matrix of A, denoted by A^{-1}, is given by:

$$A^{-1} = \begin{pmatrix} \frac{5}{4} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{4} \end{pmatrix}$$

$$\begin{pmatrix} a & b & 1 & 0 \\ c & d & 0 & 1 \end{pmatrix} \underset{\text{(i)}}{\Longrightarrow} \begin{pmatrix} 1 & \frac{b}{a} & \frac{1}{a} & 0 \\ c & d & 0 & 1 \end{pmatrix} \underset{\text{(ii)}}{\Longrightarrow} \begin{pmatrix} 1 & \frac{b}{a} & \frac{1}{a} & 0 \\ 0 & d - \frac{bc}{a} & -\frac{c}{a} & 1 \end{pmatrix}$$

$$\underset{\text{(iii)}}{\Longrightarrow} \begin{pmatrix} 1 & \frac{b}{a} & \frac{1}{a} & 0 \\ 0 & 1 & -\frac{c}{ad-bc} & \frac{a}{ad-bc} \end{pmatrix} \underset{\text{(iv)}}{\Longrightarrow} \begin{pmatrix} 1 & 0 & \frac{d}{ad-bc} & -\frac{b}{ad-bc} \\ 0 & 1 & -\frac{c}{ad-bc} & \frac{a}{ad-bc} \end{pmatrix}$$

(i) Divide the 1st line by a.

(ii) Subtract the 1st line times c from the 2nd line.

(iii) Divide the 2nd line times $\dfrac{a}{ad-bc}$.

(iv) Subtract the 2nd line times $\dfrac{b}{a}$ from the 1st line.

A: $n \times k$ $\qquad$ b: $k \times 1$

$$Ab = \begin{pmatrix} \sum_{j=1}^{k} a_{1j} b_j \\ \sum_{j=1}^{k} a_{2j} b_j \\ \vdots \\ \sum_{j=1}^{k} a_{kj} b_j \end{pmatrix}, \quad \text{i.e.,} \quad \text{the } i\text{th element of } Ab \text{ is } \sum_{j=1}^{k} a_{ij} b_j$$

Matrix × Vector

```
#include<stdio.h>
#include<math.h>

void main()
{
  int i,j,k;
  double x[10],a[10][10],b[10];

  a[1][1]=1.0;
  a[1][2]=1.0;
```

```
    a[2][1]=a[1][2];
    a[2][2]=5.0;

    b[1]=1.0;
    b[2]=2.0;

    k=2;

    for(i=1;i<=k;++i){
      x[i]=0.0;
      for(j=1;j<=k;++j){
        x[i]+=a[i][j]*b[j];
      }
      printf("%10.3lf\n",x[i]);
    }

}
```

A: $n \times k$ $\qquad$ B: $k \times m$

$$AB = \begin{pmatrix} \sum_{L=1}^{k} a_{1L}b_{L1} & \sum_{L=1}^{k} a_{1L}b_{L2} & \cdots & \sum_{L=1}^{k} a_{1L}b_{Lm} \\ \sum_{L=1}^{k} a_{2L}b_{L1} & \sum_{L=1}^{k} a_{2L}b_{L2} & \cdots & \sum_{L=1}^{k} a_{2L}b_{Lm} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{L=1}^{k} a_{nL}b_{L1} & \sum_{L=1}^{k} a_{nL}b_{L2} & \cdots & \sum_{L=1}^{k} a_{nL}b_{Lm} \end{pmatrix},$$

i.e., The (i, j)th element of AB is $\sum_{L=1}^{k} a_{iL}b_{Lj} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ik}b_{kj}$

OLS

```
#include<stdio.h>
#include<math.h>

double xx[100][100];

void main()
{
    double x[1000][100],y[1000],b[100],xy[100];
    int i,j,k,m,n;
    void inverse(int k);

/*
   input: x[1000][100],y[1000],n,k
   output: b[100]
*/

    n=5;
    k=2;
```

```
    FILE *fn;
    fn=fopen("data.txt","r");
    for(m=1;m<=n;++m){
      fscanf(fn,"%lf",&y[m]);
      for(j=1;j<=k;++j) fscanf(fn,"%lf",&x[m][j]);
      if(m != n) fscanf(fn,"\n");
    }
    fclose(fn);

    for(i=1;i<=k;++i){
      xy[i]=0.0;
      for(m=1;m<=n;++m) xy[i]+=x[m][i]*y[m];
    }
    for(i=1;i<=k;++i){
      for(j=1;j<=k;++j){
        xx[i][j]=0.0;
        for(m=1;m<=n;++m) xx[i][j]+=x[m][i]*x[m][j];
      }
    }
```

```

    inverse(k);
    for(i=1;i<=k;++i){
      b[i]=0.0;
      for(j=1;j<=k;++j) b[i]+=xx[i][j]*xy[j];
    }

    printf("Regression Coeff.:\n");
    for(i=1;i<=k;++i) printf("%15.7lf",b[i]);

    fn=fopen("res.txt","w");
    fprintf(fn,"Regression Coeff.:\n");
    for(i=1;i<=k;++i) fprintf(fn,"%15.7lf",b[i]);
    fclose(fn);

}
/* --------------------------------------------------------- */
void inverse(int k)
{
   double a1,a2;
```

```
   int    i,j,m;

   for(i=1;i<=k;i++){
      a1=xx[i][i];
      xx[i][i]=1.0;
      for(j=1;j<=k;j++) xx[i][j]=xx[i][j]/a1;
      for(m=1;m<=k;m++){
         if(m != i){
            a2=xx[m][i];
            xx[m][i]=0.0;
            for(j=1;j<=k;j++) xx[m][j]-=a2*xx[i][j];
         }
      }
   }
}
```

For the regression model: $y = X\beta + u$, the OLS estimator is given by $\hat{\beta} = (X'X)^{-1}X'y$.

y: $n \times 1$, X: $n \times k$, β: $k \times 1$, u: $n \times 1$

- In Lines 20 – 27, y and X are input from the text file `data.txt`, which are stored as `y[m]` in Line 23 and `x[m][j]` in Line 24. `fn` in Lines 20, 21, 23 – 25, 27 should be the same name.
- In Lines 29 – 32, $X'y$ is computed as `xy[i]`.
- In Lines 33 – 38, $X'X$ is computed as `xx[i][j]`.
- In Line 40, $(X'X)^{-1}$ is computed as `xx[i][j]`. That is, `xx[i][j]` is overwritten from $X'X$ to $(X'X)^{-1}$ in Line 40.
- In Lines 41 – 44, $(X'X)^{-1}X'y$ is computed by multiplying `xx[i][j]` by `xy[i]`, which is $\hat{\beta}$, stored in `b[i]`.
- In Lines 46 and 47, the OLS estimate `b[i]` is shown in the screen.
- In Lines 20 and 49 – 52, the OLS estimate `b[i]` is stored in the text file `res.txt`. `fn` in Lines 20, 49 – 52 should be the same name.
- `"r"` in Line 21 indicates "read", and `"w"` in Line 49 indicates "write".