Intermediate C Programming

Lesson4

Pointers and Matrices

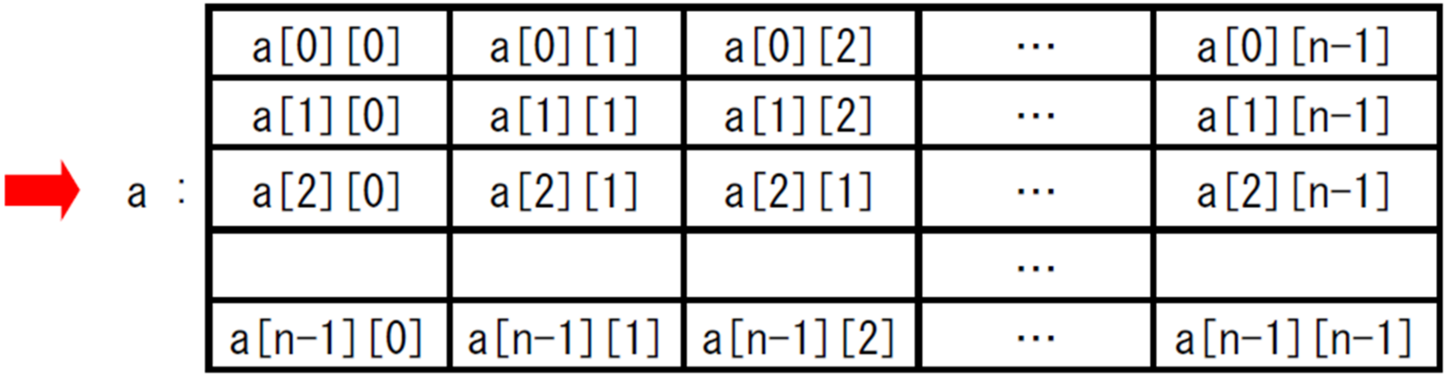
Today's outline

- Pointers and Matrices
- Exercise

Pointers and Matrices

- If we define

double a[n][n];



a :

a[0][0]	a[0][1]	a[0][2]	…	a[0][n-1]
a[1][0]	a[1][1]	a[1][2]	…	a[1][n-1]
a[2][0]	a[2][1]	a[2][1]	…	a[2][n-1]
			…	
a[n-1][0]	a[n-1][1]	a[n-1][2]	…	a[n-1][n-1]

Pointers and Matrices

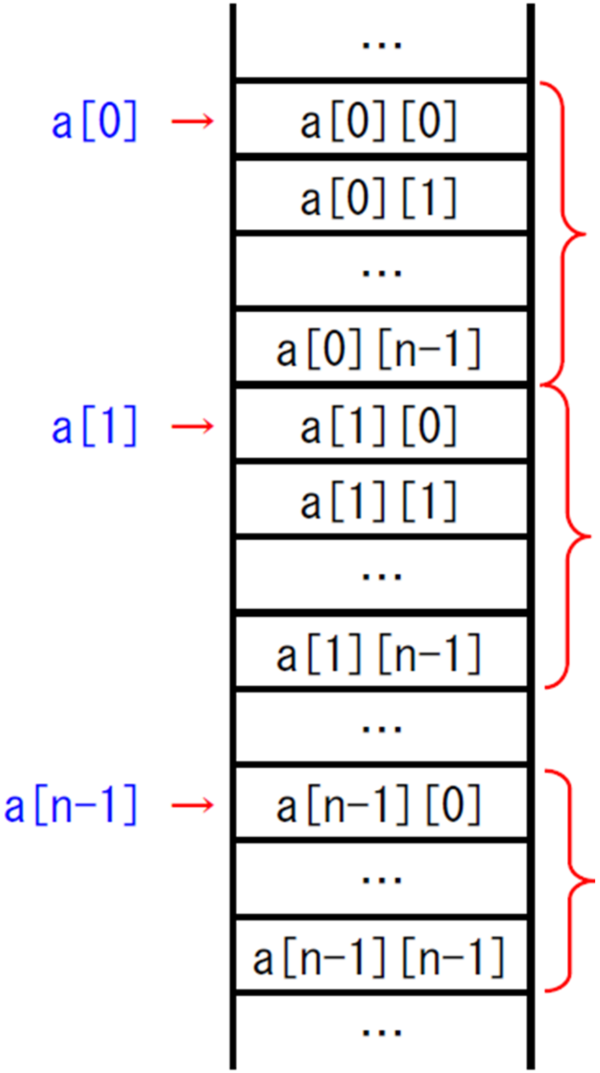
- If we define

  double a[n][n];

  a[0], a[1], ··· , a[n−1]

- The pointer of 1st element of i-line

  a[i] (i=0, 1, ···, n−1)



···

a[0] → a[0][0]

a[0][1]

···

a[0][n−1]

a[1] → a[1][0]

a[1][1]

···

a[1][n−1]

···

a[n−1] → a[n−1][0]

···

a[n−1][n−1]

···

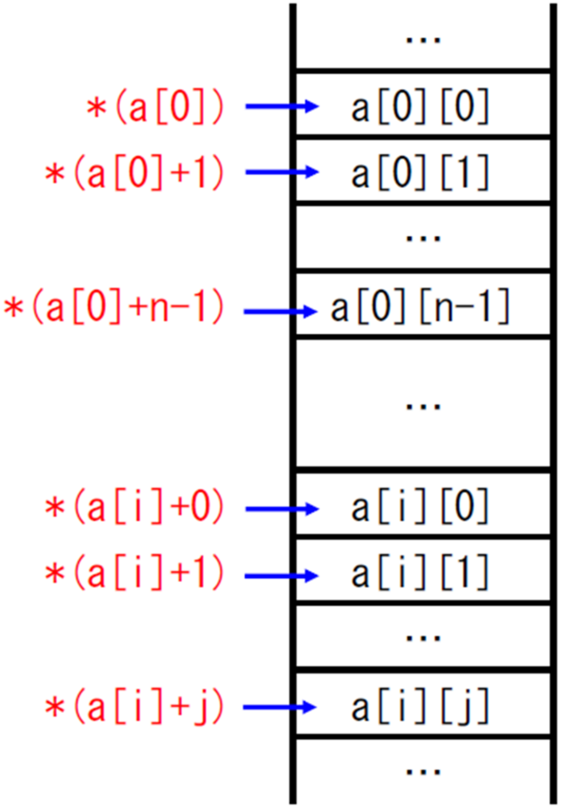
Pointers and Matrices

- The j'th element of i-line

  *(a[i] + j)

- The j'th element of i-line

  a[i][j] ⇔ *(a[i] + j)



Pointers and Matrices

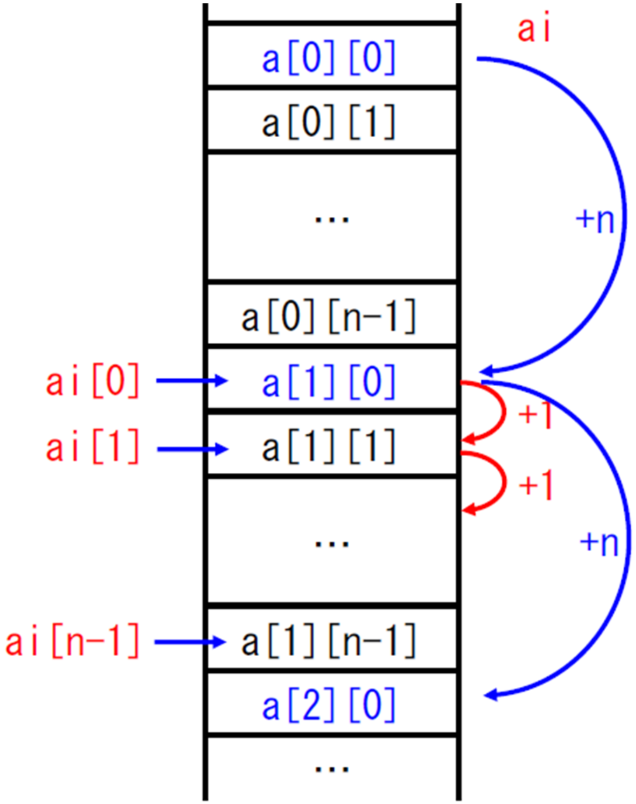
- If we define

  double *ai;

  ai = a;

- ai will be the pointer of the 1st element of the 1st line
- ai = a=&a[0][0]

- ai += n
- ai will be the pointer of the 1st element of the next line

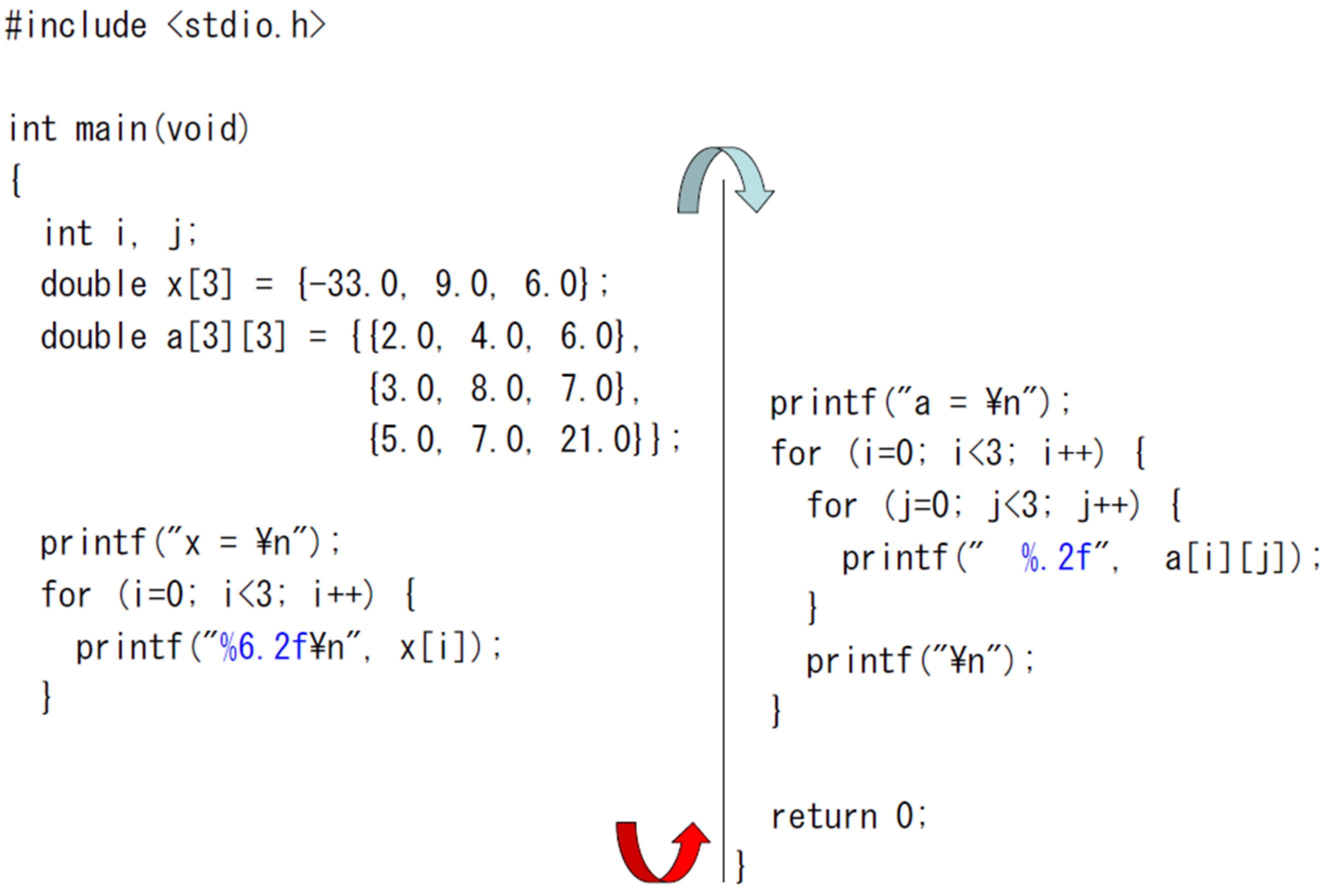


Exercise

$$A = \begin{pmatrix} 2 & 4 & 6 \\ 3 & 8 & 7 \\ 5 & 7 & 21 \end{pmatrix}, \quad x = \begin{pmatrix} -33 \\ 9 \\ 6 \end{pmatrix}$$

Exercise

- Program to show A and x



```
#include <stdio.h>

int main(void)
{
  int i, j;
  double x[3] = {-33.0, 9.0, 6.0};
  double a[3][3] = {{2.0, 4.0, 6.0},
                    {3.0, 8.0, 7.0},
                    {5.0, 7.0, 21.0}};

  printf(”x = ¥n”);
  for (i=0; i<3; i++) {
    printf(”%6.2f¥n”, x[i]);
  }
  printf(”a = ¥n”);
  for (i=0; i<3; i++) {
    for (j=0; j<3; j++) {
      printf(” %.2f”, a[i][j]);
    }
    printf(”¥n”);
  }

  return 0;
}
```

Exercise

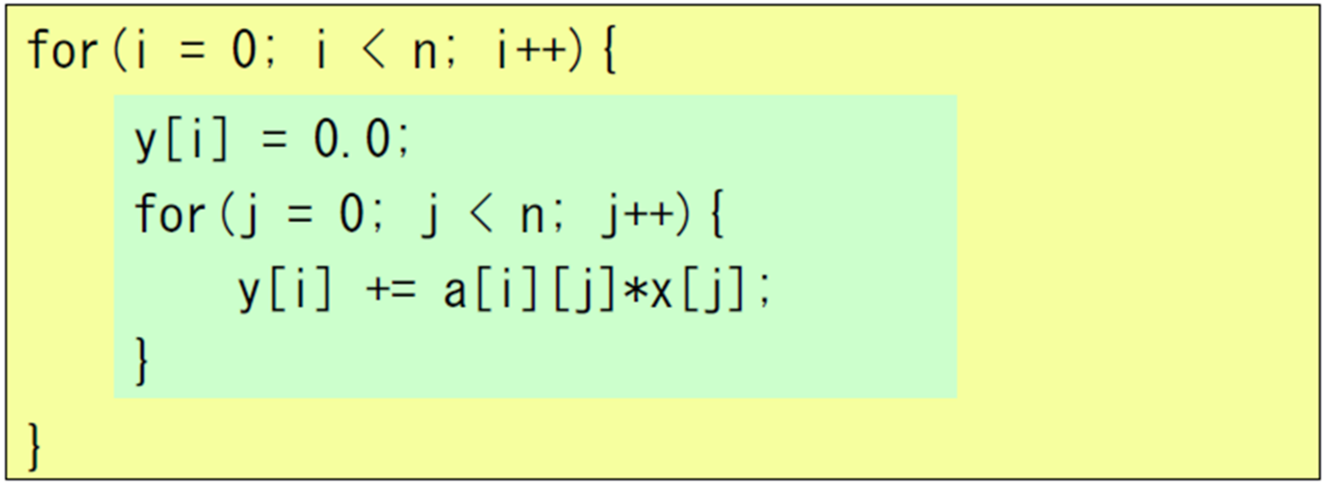
$$y = A \cdot x$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} & \cdots & a_{0,n-1} \\ a_{10} & a_{11} & \cdots & a_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n-1,0} & a_{n-1,1} & \cdots & a_{n-1,n-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix}$$

$$y_i = a_{i0}x_0 + a_{i1}x_1 + \cdots + a_{i,n-1}x_{n-1} \quad (i = 0,\ 1,\ \cdots,\ n-1)$$

```
y[i] = 0.0;
for(j = 0; j < n; j++){
    y[i] += a[i][j]*x[j];
}
```

Exercise



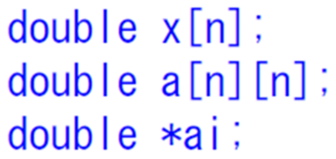
```
for(i = 0; i < n; i++){
    y[i] = 0.0;
    for(j = 0; j < n; j++){
        y[i] += a[i][j]*x[j];
    }
}
```

Exercise

- See source code
- mul_mv_nopointer.c

Exercise

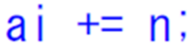
- If we declare



```
double x[n];
double a[n][n];
double *ai;
```

- Each element of matrix

- ai becomes the pointer of the 1st element of the next line



```
ai += n;
```

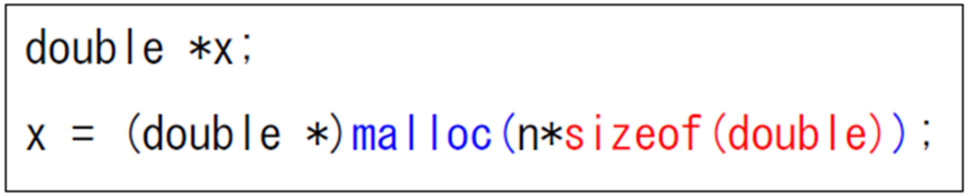
Exercise

- For

$$A = \begin{pmatrix} 2 & 4 & 6 \\ 3 & 8 & 7 \\ 5 & 7 & 21 \end{pmatrix}, \quad x = \begin{pmatrix} -33 \\ 9 \\ 6 \end{pmatrix}$$

- How to write the program to calculate $(A \cdot x)$ with pointers?

malloc(size)

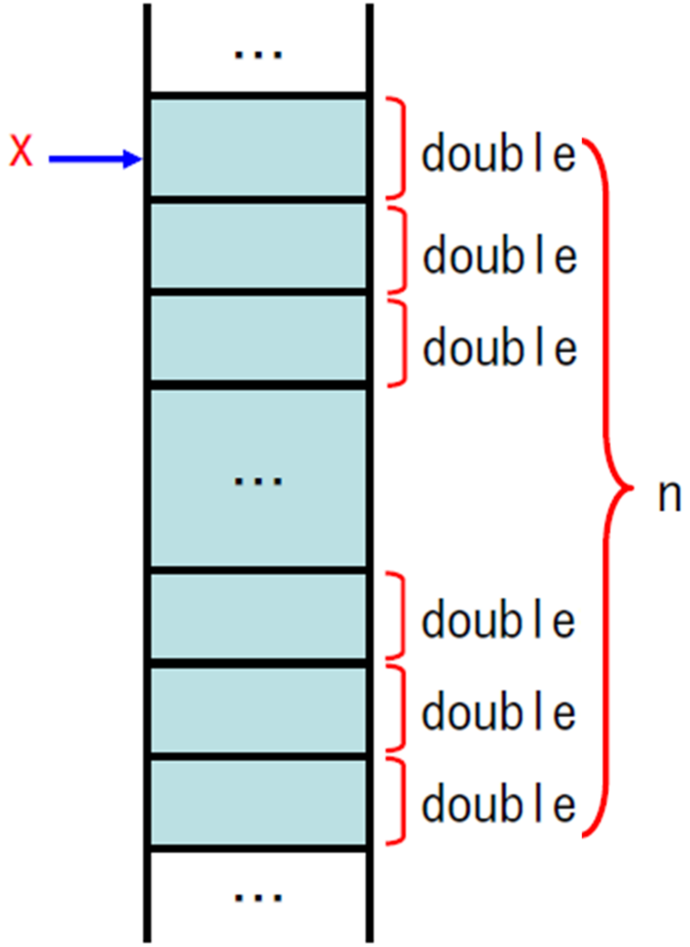


```
double *x;

x = (double *)malloc(n*sizeof(double));
```

sizeof(double) → 8 byte

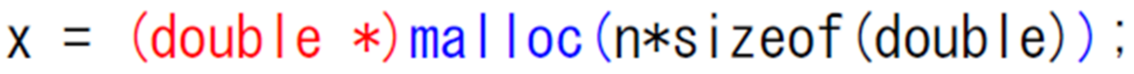
sizeof(int) → 4 byte

sizeof(char) → 1 byte



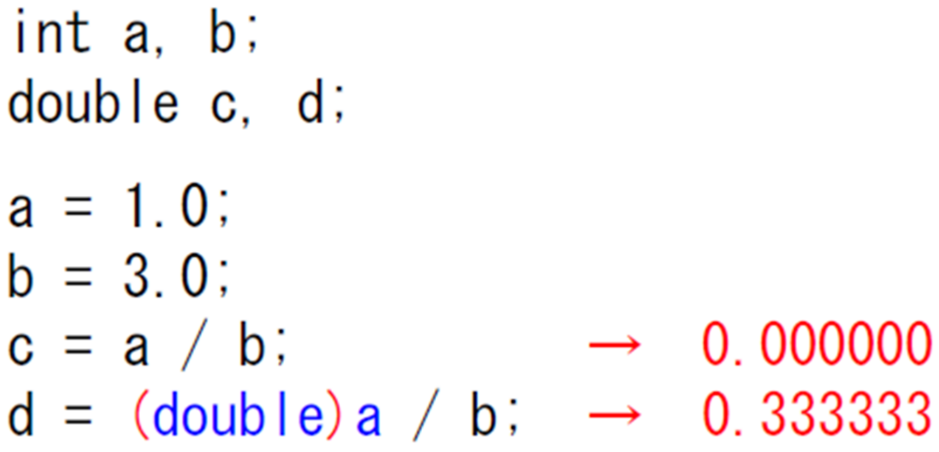
malloc(size)

- x



```
x = (double *)malloc(n*sizeof(double));
```

- (double)



```
int a, b;
double c, d;

a = 1.0;
b = 3.0;
c = a / b;             →  0.000000
d = (double)a / b;     →  0.333333
```

malloc(size)

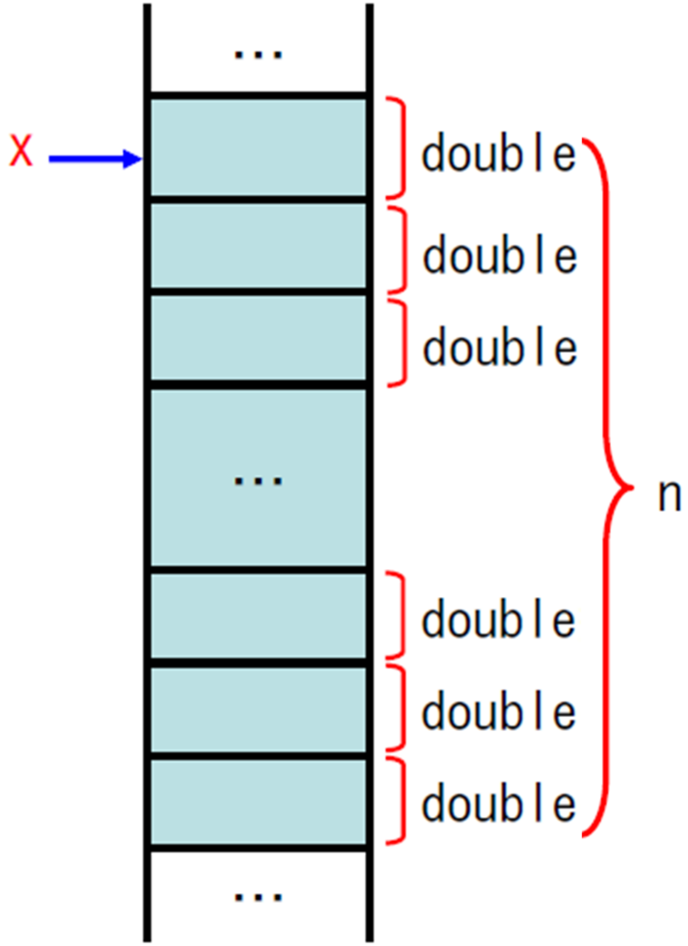
```
double *x;

x = (double *)malloc(n*sizeof(double));
```

- x + i — Pointer to the i'th element
- *(x + i) — The i'th element

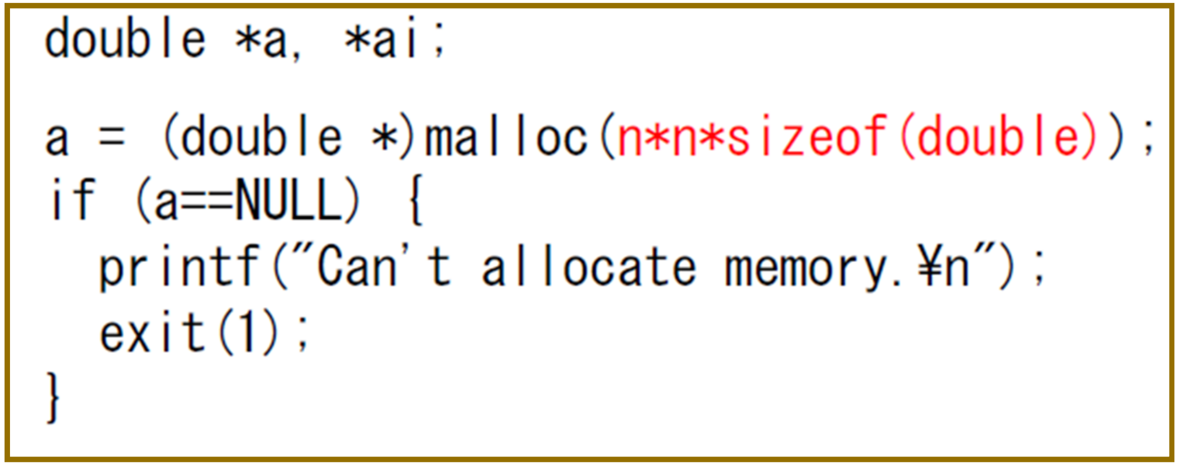
↕

x[i]

```
scanf("%lf", &x[i]);      →
printf("%.2f¥n", x[i]);   →
```



malloc(size)



```
double *a, *ai;

a = (double *)malloc(n*n*sizeof(double));
if (a==NULL) {
  printf("Can't allocate memory.¥n");
  exit(1);
}
```

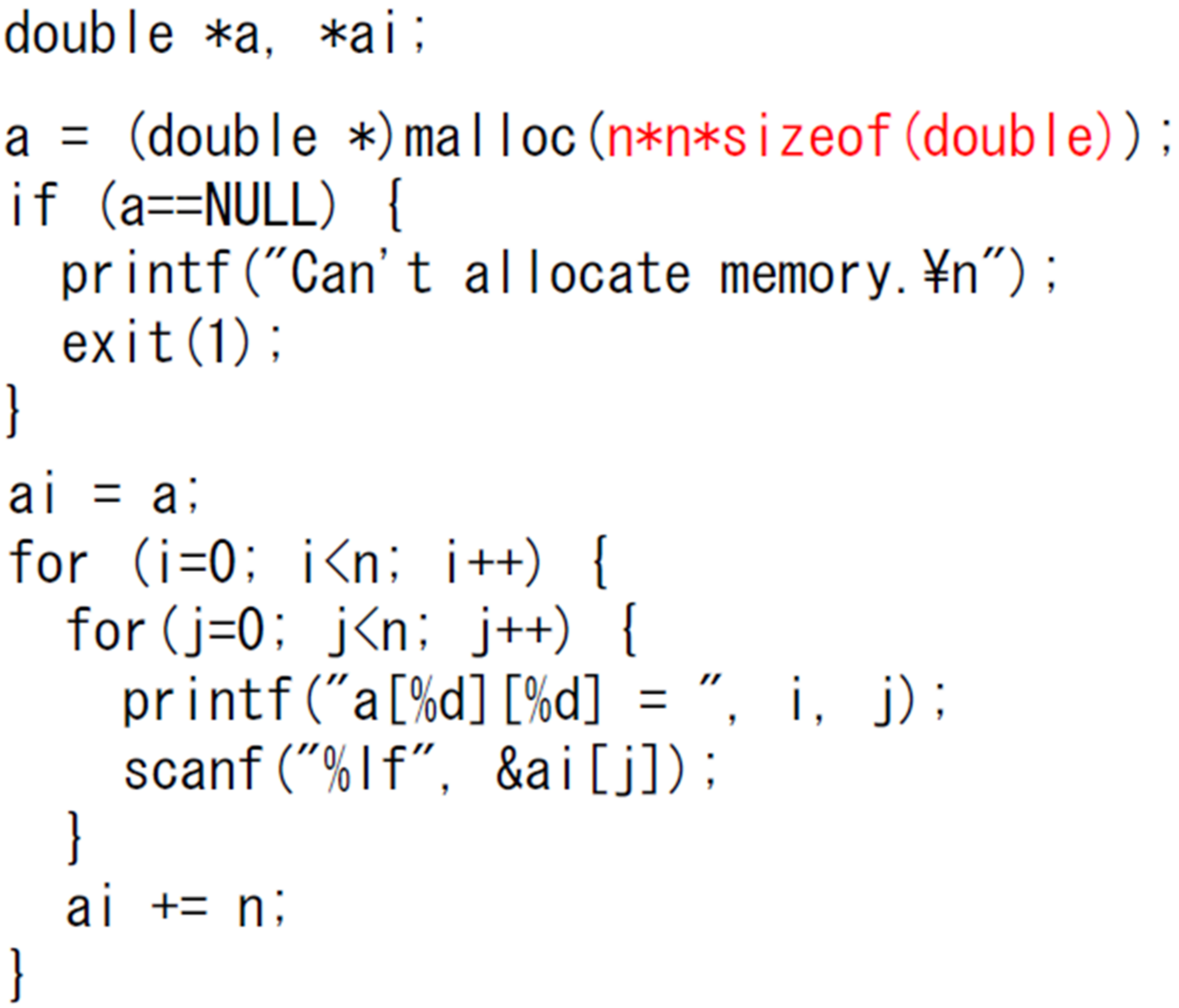
- Poniter ai

```
ai = a;
```

- Change line

```
ai += n;
```

malloc(size)



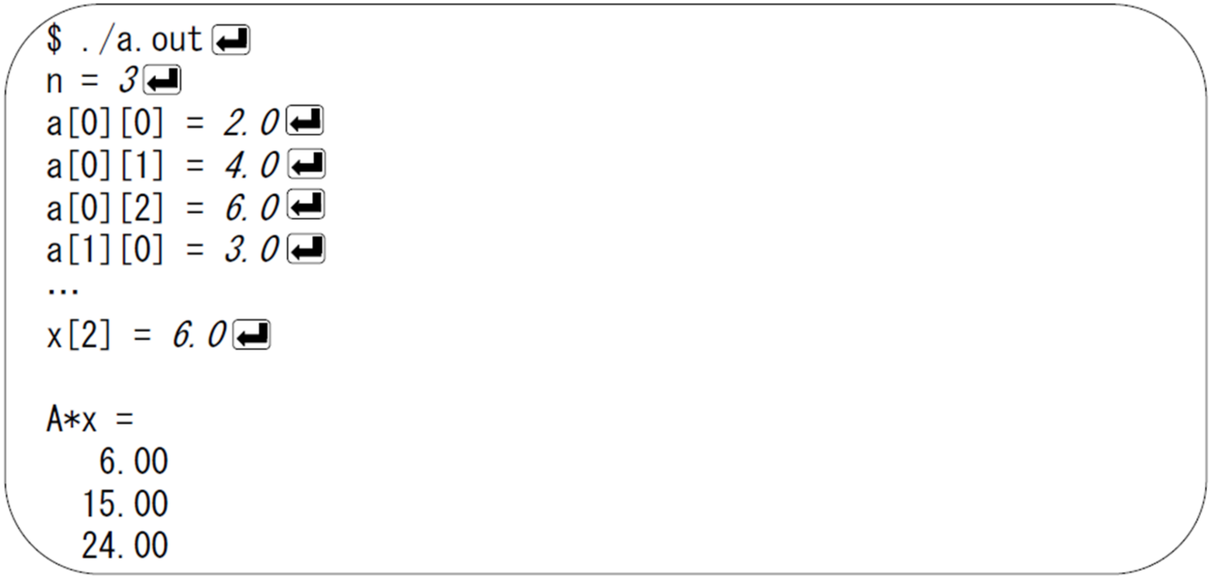
```
double *a, *ai;

a = (double *)malloc(n*n*sizeof(double));
if (a==NULL) {
  printf(”Can’t allocate memory.¥n”);
  exit(1);
}

ai = a;
for (i=0; i<n; i++) {
  for(j=0; j<n; j++) {
    printf(”a[%d][%d] = ”, i, j);
    scanf(”%lf”, &ai[j]);
  }
  ai += n;
}
```

Exercise

- For

$$A = \begin{pmatrix} 2 & 4 & 6 \\ 3 & 8 & 7 \\ 5 & 7 & 21 \end{pmatrix}, \quad x = \begin{pmatrix} -33 \\ 9 \\ 6 \end{pmatrix}$$

- How to write the program to calculate （$A \cdot x$） with pointers and malloc(size)?



```
$ ./a.out ↵
n = 3↵
a[0][0] = 2.0↵
a[0][1] = 4.0↵
a[0][2] = 6.0↵
a[1][0] = 3.0↵
…
x[2] = 6.0↵

A*x =
   6.00
  15.00
  24.00
```

Homework

- For

$$A=\begin{pmatrix}3 & -1 & 2\\ 1 & 2 & 3\\ 2 & -2 & -1\end{pmatrix},\quad B=\begin{pmatrix}8 & 1 & -1\\ -1 & 7 & -2\\ 2 & 1 & 9\end{pmatrix}$$

- How to write the program to calculate A*B with pointers and malloc(size)?