

```
int A[M][N];
```

Memory:

	N				
M	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][2]	...	A[2][N-1]
	⋮	⋮	⋮	⋮	⋮
	A[M-1][0]	A[M-1][1]	A[M-1][2]	...	A[M-1][N-1]

Memory, linear view:

M*N									
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]

Array versus pointer: $A[i][j] = 17;$

```
int *pA = (int *)A;  
pA[i*N + j] = 17;
```

Unspecified Row Count: Ok

```
int f(int A[][N])
```

Memory:

	N				
	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][2]	...	A[2][N-1]
	⋮	⋮	⋮	⋮	⋮

Array versus pointer: `A[i][j] = 17;`

```
int *pA = (int *)A;  
pA[i*N + j] = 17;
```

Unspecified Column Count: Not Ok

```
int f(int A[M][ ])
```

Memory:

	?			
Σ	A[0][0]	A[0][1]	A[0][2]	...
	A[1][0]	A[1][1]	A[1][2]	...
	A[2][0]	A[2][1]	A[2][2]	...
	⋮	⋮	⋮	⋮
	A[M-1][0]	A[M-1][1]	A[M-1][2]	...

Array versus pointer: $A[i][j] = 17;$

```
int *pA = (int *)A;
```

```
pA[i*? + j] = 17;
```

```
#define M ??  
#define N ??  
  
int mat1[M][N];  
int mat2[N][M];  
  
int sum_element(int i, int j) {  
    return mat1[i][j] + mat2[j][i];  
}
```

```
movl 8(%ebp),%ecx  
movl 12(%ebp),%eax  
leal 0(,%eax,4),%ebx  
leal 0(,%ecx,8),%edx  
subl %ecx,%edx  
addl %ebx,%eax  
sall $2,%eax  
movl mat2(%eax,%ecx,4),%eax  
addl mat1(%ebx,%edx,4),%eax
```

```

#define M ??
#define N ??

int mat1[M][N], pmat1 = (int*)mat1;
int mat2[N][M], pmat2 = (int*)mat2;

int sum_element(int i, int j) {
    return mat1[i][j] + mat2[j][i];
}

```

```

mat1[i][j]
pmat1[i*N+j]
pmat1 @ (i*N+j)*4
N = 7
pmat1 @ (i*7+j)*4
pmat1 @ (i7+j)*4
pmat1 @ i7*4+j4

```

movl 8(%ebp), %ecx	i
movl 12(%ebp), %eax	j
leal 0(, %eax, 4), %ebx	j4 = 4*j
leal 0(, %ecx, 8), %edx	i8 = i*8
subl %ecx, %edx	i7 = i8-i
addl %ebx, %eax	j5 = j4+j
sall \$2, %eax	j20 = 4*j5
movl mat2(%eax, %ecx, 4), %eax	pmat2 @ j20+4*i
addl mat1(%ebx, %edx, 4), %eax	pmat1 @ i7*4+j4

```

#define M ??
#define N ??

int mat1[M][N], pmat1 = (int*)mat1;
int mat2[N][M], pmat2 = (int*)mat2;

int sum_element(int i, int j) {
    return mat1[i][j] + mat2[j][i];
}

```

```

mat2[j][i]
pmat2[j*M+i]
pmat2 @ (j*M+i)*4
M = 5
pmat2 @ (j*5+i)*4
pmat2 @ j20+4*i

```

movl 8(%ebp), %ecx	i
movl 12(%ebp), %eax	j
leal 0(, %eax, 4), %ebx	$j4 = 4*j$
leal 0(, %ecx, 8), %edx	$i8 = i*8$
subl %ecx, %edx	$i7 = i8 - i$
addl %ebx, %eax	$j5 = j4 + j$
sall \$2, %eax	$j20 = 4*j5$
movl mat2(%eax, %ecx, 4), %eax	$pmat2 @ j20 + 4*i$
addl mat1(%ebx, %edx, 4), %eax	$pmat1 @ i7*4 + j4$