



Lecture 15

Arrays

CSE115: Computing Concepts

Two-dimensional Arrays

- A two-dimensional array consists of both rows and columns of elements. It is essentially a matrix.
- To declare a two-dimensional array, we merely use two sets of square brackets.
 - The first contains the number of rows
 - The second contains the number of columns

```
//Creates a 2D array with 3 rows and 4 columns  
int values[3][4];
```

Indices in 2D Arrays

- Assume that the two dimensional array called **nums** is declared and looks like the following:

nums	Col 0	Col 1	Col 2	Col 3
Row 0	8	16	9	52
Row 1	3	15	27	6
Row 2	14	25	2	10

- To access the cell containing 6, we use **nums[1][3]**, that is, row 1, column 3 of the array **nums**.

Using 2D Arrays

- Just like 1D arrays, once you have specified the index, you are just working with a single variable of the given data type.
- Assignments and usage is still the same:

```
sumRow0 = nums[0][0] + nums[0][1] + nums[0][2] +  
          nums[0][3];
```

```
//assigns 72 to cell at row 2, column 3  
nums[2][3] = 72;
```

Initializing 2D Arrays

- You can use additional braces to indicate when rows start and end, but you don't have to do that.

```
int nums[3][4] = {{8,16,9,52},  
                  {3,15,27,6},  
                  {14,25,2,10}  
                };
```

- Or

```
int nums[3][4] = {8,16,9,52,  
                  3,15,27,6,  
                  14,25,2,10};
```

- Or (correct, but not as clear as the first two):

```
int nums[3][4] = {8,16,9,52,3,15,27,6,14,25,2,10};
```

More on 2D Arrays

- Initialization of 2D arrays is done in ***row order***.
- 2D arrays work well with (for) loops like 1D arrays.
However, to access all elements, typically you will need nested loops for 2D arrays. Can you see why?

Example (Display Array Values)

```
int main()
{
    int A[3][4] = { {11, 12, 13, 14},
                    {21, 22, 23, 24},
                    {31, 32, 33, 34} };

    int i, j;

    printf("Values in array A:\n");
    for(i=0;i<3;i++)
    {
        for(j=0;j<4;j++)
        {
            printf("%10d ",A[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Example (Array Values as User-input)

```
int main()
{
    int A[100][100], i, j, rows, columns;
    printf("Number of rows: ");
    scanf("%d",&rows);
    printf("Number of columns: ");
    scanf("%d",&columns);
    for(i=0;i<rows;i++)
    {
        for(j=0;j<columns;j++)
        {
            printf("A[%d][%d]: ",i, j);
            scanf("%d",&A[i][j]);
        }
    }
    printf("Values in array A:\n");
    for(i=0;i<rows;i++)
    {
        for(j=0;j<columns;j++)
        {
            printf("%10d ",A[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```


Example (Add Two Matrices) 1/3

```
int main()
{
    int A[100][100], B[100][100], C[100][100];
    int i, j, rowsA, columnsA, rowsB, columnsB;

    printf("Number of rows in A: ");
    scanf("%d", &rowsA);
    printf("Number of columns in A: ");
    scanf("%d", &columnsA);
    printf("Number of rows in B: ");
    scanf("%d", &rowsB);
    printf("Number of columns in B: ");
    scanf("%d", &columnsB);

    if(rowsA != rowsB || columnsA != columnsB)
    {
        printf("Invalid matrix dimensions\n");
        return 0;
    }
}
```

Example (Add Two Matrices) 2/3

```
for (i=0; i<rowsA; i++)
{
    for (j=0; j<columnsA; j++)
    {
        printf("A[%d][%d]: ", i, j);
        scanf("%d", &A[i][j]);
    }
}
printf("\n");
for (i=0; i<rowsB; i++)
{
    for (j=0; j<columnsB; j++)
    {
        printf("B[%d][%d]: ", i, j);
        scanf("%d", &B[i][j]);
    }
}
```

Example (Add Two Matrices) 3/3

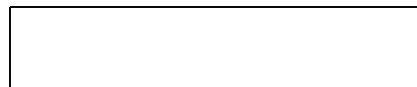
```
for (i=0; i<rowsA; i++)
{
    for (j=0; j<columnsA; j++)
    {
        C[i][j] = A[i][j] + B[i][j];
    }
}
printf("Result:\n");
for (i=0; i<rowsA; i++)
{
    for (j=0; j<columnsA; j++)
    {
        printf("%10d ", C[i][j]);
    }
    printf("\n");
}
return 0;
}
```

Example (Multiply Two Matrices)

- Multiply rows with columns.
- You can only multiply if the number of columns in the 1st matrix is equal to the number of rows in the 2nd matrix.

$$\begin{bmatrix} -3 & 2 & 5 \\ 7 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} -8 & 2 \\ 1 & 5 \\ 0 & -3 \end{bmatrix}$$

They must match.



Dimensions:

2 x 3

3 x 2



The dimensions of your answer.

Example (Multiply Two Matrices)

$$\begin{matrix} & \mathbf{A} & & \mathbf{B} \\ \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix} & \times & \begin{bmatrix} 3 & -9 & 2 \\ 5 & 7 & -6 \end{bmatrix} \end{matrix}$$

Example (Multiply Two Matrices)

A

$$\begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix}$$

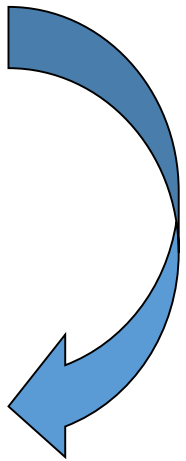
B

$$\times \begin{bmatrix} 3 & -9 & 2 \\ 5 & 7 & -6 \end{bmatrix}$$

$$\begin{bmatrix} 2(3) + -1(5) & 2(-9) + -1(7) & 2(2) + -1(-6) \\ 3(3) + 4(5) & 3(-9) + 4(7) & 3(2) + 4(-6) \end{bmatrix}$$

$$\begin{bmatrix} 1 & -25 & 10 \\ 29 & 1 & -18 \end{bmatrix}$$

C



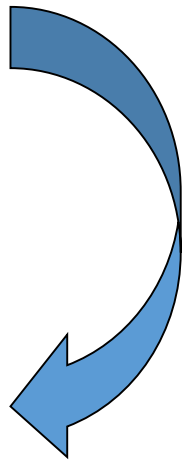
Example (Multiply Two Matrices)

$$\begin{matrix} \mathbf{A} & & \mathbf{B} \\ \begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix} & \times & \begin{bmatrix} 3 & -9 & 2 \\ 5 & 7 & -6 \end{bmatrix} \end{matrix}$$

$$\begin{bmatrix} 2(3) + -1(5) & 2(-9) + -1(7) & 2(2) + -1(-6) \\ 3(3) + 4(5) & 3(-9) + 4(7) & 3(2) + 4(-6) \end{bmatrix}$$

$$\begin{bmatrix} 1 & -25 & 10 \\ 29 & 1 & -18 \end{bmatrix}$$

C



Example (Multiply Two Matrices)

A	Col 0	Col 1	×	B	Col 0	Col 1	Col 2
Row 0	2	-1		Row 0	3	-9	2
Row 1	3	4		Row 1	5	7	-6

$$\begin{bmatrix} 2(3) + -1(5) & 2(-9) + -1(7) & 2(2) + -1(-6) \\ 3(3) + 4(5) & 3(-9) + 4(7) & 3(2) + 4(-6) \end{bmatrix}$$

```
sum = 0;
for(k=0;k<columnsA;k++)
    sum += A[0][k]*B[k][0];
C[0][0] = sum;
```

$$\begin{bmatrix} 1 & -25 & 10 \\ 29 & 1 & -18 \end{bmatrix}$$

C

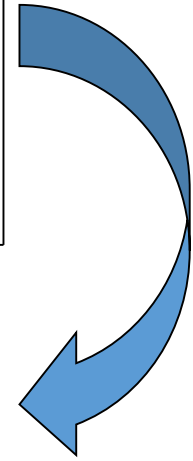
Example (Multiply Two Matrices)

A $\begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix} \times$ **B** $\begin{bmatrix} 3 & -9 & 2 \\ 5 & 7 & -6 \end{bmatrix}$

$$\begin{bmatrix} 2(3) + -1(5) & 2(-9) + -1(7) & 2(2) + -1(-6) \\ 3(3) + 4(5) & 3(-9) + 4(7) & 3(2) + 4(-6) \end{bmatrix}$$

$$\begin{bmatrix} 1 & -25 & 10 \\ 29 & 1 & -18 \end{bmatrix}$$

C



Example (Multiply Two Matrices)

A	Col 0	Col 1
Row 0	2	-1
Row 1	3	4

 ×

B	Col 0	Col 1	Col 2
Row 0	3	-9	2
Row 1	5	7	-6

$$\begin{bmatrix} 2(3) + -1(5) & 2(-9) + -1(7) & 2(2) + -1(-6) \\ 3(3) + 4(5) & 3(-9) + 4(7) & 3(2) + 4(-6) \end{bmatrix}$$

```
for (j=0; j<columnsB; j++)  
{
```

```
    sum = 0;
```

```
    for (k=0; k<columnsA; k++)
```

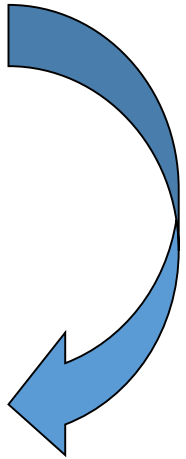
```
        sum += A[0][k]*B[k][j];
```

```
    C[0][j] = sum;
```

```
}
```

$$\begin{bmatrix} 1 & -25 & 10 \\ 29 & 1 & -18 \end{bmatrix}$$

C



Example (Multiply Two Matrices)

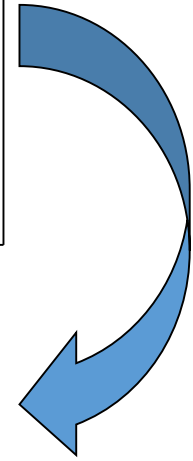
A

$$\begin{bmatrix} 2 & -1 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 3 & -9 & 2 \\ 5 & 7 & -6 \end{bmatrix}

B

$$\begin{bmatrix} 2(3) + -1(5) & 2(-9) + -1(7) & 2(2) + -1(-6) \\ 3(3) + 4(5) & 3(-9) + 4(7) & 3(2) + 4(-6) \end{bmatrix}$$

C

$$\begin{bmatrix} 1 & -25 & 10 \\ 29 & 1 & -18 \end{bmatrix}$$
$$

Example (Multiply Two Matrices)

A	Col 0	Col 1
Row 0	2	-1
Row 1	3	4

 ×

B	Col 0	Col 1	Col 2
Row 0	3	-9	2
Row 1	5	7	-6

$$\begin{bmatrix} 2(3) + -1(5) & 2(-9) + -1(7) & 2(2) + -1(-6) \\ 3(3) + 4(5) & 3(-9) + 4(7) & 3(2) + 4(-6) \end{bmatrix}$$

```
for(i=0;i<rowsA;i++)
```

```
{
```

```
    for(j=0;j<columnsB;j++)
```

```
    {
```

```
        sum = 0;
```

```
        for(k=0;k<columnsA;k++)
```

```
            sum += A[i][k]*B[k][j];
```

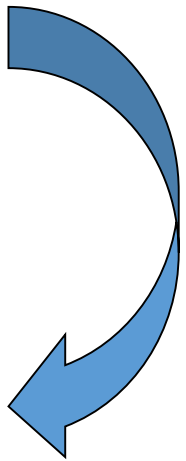
```
        C[i][j] = sum;
```

```
    }
```

```
}
```

$$\begin{bmatrix} 1 & -25 & 10 \\ 29 & 1 & -18 \end{bmatrix}$$

C



Example (Multiply Two Matrices) 1/3

```
int main()
{
    int A[100][100], B[100][100], C[100][100];
    int i, j, k, rowsA, columnsA, rowsB, columnsB;
    int sum;

    printf("Number of rows in A: ");
    scanf("%d", &rowsA);
    printf("Number of columns in A: ");
    scanf("%d", &columnsA);
    printf("Number of rows in B: ");
    scanf("%d", &rowsB);
    printf("Number of columns in B: ");
    scanf("%d", &columnsB);

    if(columnsA != rowsB)
    {
        printf("Invalid matrix dimensions\n");
        return 0;
    }
}
```

Example (Multiply Two Matrices) 2/3

```
for (i=0; i<rowsA; i++)
{
    for (j=0; j<columnsA; j++)
    {
        printf("A[%d][%d]: ", i, j);
        scanf("%d", &A[i][j]);
    }
}
printf("\n");
for (i=0; i<rowsB; i++)
{
    for (j=0; j<columnsB; j++)
    {
        printf("B[%d][%d]: ", i, j);
        scanf("%d", &B[i][j]);
    }
}
```

Example (Multiply Two Matrices) 3/3

```
for (i=0; i<rowsA; i++)
{
    for (j=0; j<columnsB; j++)
    {
        sum = 0;
        for (k=0; k<columnsA; k++)
            sum += A[i][k]*B[k][j];
        C[i][j] = sum;
    }
}
printf("Result:\n");
for (i=0; i<rowsA; i++)
{
    for (j=0; j<columnsB; j++)
    {
        printf("%10d ", C[i][j]);
    }
    printf("\n");
}
return 0;
}
```