Arrays (2)
Higher-Dimensional Arrays
Arrays of Character Strings
Topics

• Variables and Arrays (review)
• One-Dimensional Arrays (review)
• High-Dimensional Arrays
  • Declaration of higher dimensional arrays
  • Accessing elements of high-dimensional arrays
  • Initializing high-dimensional arrays
  • Using loops with high-dimensional arrays
• Arrays of Character Strings
• Examples
Variables and Arrays (review)

- A *variable* is a location to store a value
- A *variable* is defined by its
  - Name (num, total, etc.)
  - Data type (int, float, char)
- A variable can store only one data item
- Example:
  ```
  int k;
  k=15;
  ```
Variables and Arrays (review)

- If you want to store the grade of one student, you will need a variable such as
  - float grade;

- If you want to store several grades (20, 100, 1000, ...) you will need more variables (20, 100, 1000, ...)

- **Problem:** Declaring and using this many variables is difficult.

- **Solution:** Arrays
Arrays (review)

• An array is a group of variables:
  • With a common name
  • With the same data types

• Syntax of array declaration is:
  • `DataType ArrayName [ Number of Elements];`
  • Example : `int list[25];`
Two-Dimensional Arrays

- A one-dimensional array is a row or column of variables.

```java
int A[5];
```

- A two-dimensional array is a matrix of variables.
Higher-Dimensional Arrays

- It is possible to have higher dimensional arrays too.
- For instance, a three-dimensional array is a cube of variables.
- Example: A 3x4x2 cube of variables (three dimensional array)
  
  (Note: Assume each box as a variable)
Declaring Arrays

- Declaring arrays:
  - Same as other variables, arrays should be declared
  - In declaring an array
    - Array name
    - Data type
    - Number of elements at each dimension
      Should be given.

- Syntax of declaring two-dimensional arrays is:
  - `DataType ArrayName [Number of Rows][Number of Columns];`
  - Example: `int list[5][4];`
    creates a matrix of integer variables with 5 rows and 4 columns named `list`
Example

- int A[3][5];
- float B[2][3];
- char Name[2][10];
Initializing Arrays

• We can give initial values to arrays in their declaration.

• Syntax:
  
  ```
  DataType
  ArrayName[rows][columns]=\{\{row1\},\{row2\},..\};
  ```

Example:

```
int A[2][4]=\{\{1,6,8,15\},\{7, 5, 2, 11\}\};
```

![Array Initialization Example](image)
Accessing Array Elements (1)

- To assign a value to an array element, or to use the value of an array element, we use indexes.

- Indexes should show the location of the variable (row, column, etc.)

- int A[8][5];
- A[3][4] = 0; At 4\textsuperscript{th} row and 5\textsuperscript{th} column store zero
Accessing Array Elements (2)

- Example: (In C indexes start from zero.)

  ```
  int A[2][4];
  A[1][3] = 9;
  A[0][1] = 33;
  A[0][0] = 8;
  ```
Using Two-Dimensional Arrays in \texttt{scanf/printf}

- To read a value into a variable at row \( r \) and column \( c \) of a two-dimensional array use:
  - \&\texttt{Array\_name}[r][c] with \texttt{scanf}

- To print a value of a variable at row \( r \) and column \( c \) of a two-dimensional array use:
  - \texttt{Array\_name}[r][c] with \texttt{printf} \textbf{without \& sign}
Example 1

- Write a program to create a two-dimensional array of float variables. The array should have 3 rows and 2 columns. Then read values into these variables using `scanf`.

- Print the values of the first column (column 0).
```c
#include<stdio.h>
void main()
{
    float A[3][2];

    scanf("%f%f", &A[0][0], &A[0][1]);    row 0
    scanf("%f%f", &A[1][0], &A[1][1]);    row 1
    scanf("%f%f", &A[2][0], &A[2][1]);    row 2

    /* printing the first column */
    printf("%f  %f  %f\n", A[0][0], A[1][0], A[2][0]);    First column is column zero.
    So all second indexes are 0
}
```
Using Loops with Arrays

- If you want to do the same operation with all elements of an array, you can use loops.
- For example,
  - Read values into all elements of an array,
  - Print all elements of an array
  - Add one to all elements of an array
  - Etc.
- If the array has 2 dimensions, two loops with two counters are used as indexes of the array
Example 2

- Write a program to create a two-dimensional integer array of 10 rows and 5 columns. Read values into the array. Then find sum of the values in each row of the array and print them.

- Solution:
  - Use two for loops. At each row (loop over rows) read values (loop over columns).
  - To find sum of the numbers we use a loop. At each row find the sum (second loop)
#include<stdio.h>
void main()
{
    int A[10][5], row, col;
    int sum;

    for( row = 0 ; row < 10 ; row ++ )    // for each row
        for( col = 0 ; col < 5 ; col ++ )   // At the current row, for each column
            scanf("%d", &A[row][col] );

    for( row = 0 ; row < 10; row ++ )
    {
        sum =0;
        for( col = 0 ; col < 5 ; col ++ )
            sum = sum + A[row][col];
        printf("Sum of values at row  %d  is %d \n", row, sum );
    }
}
Example 3

- Write a program to create 2 two-dimensional integer arrays of 5 rows and 5 columns. Read values into these arrays. Then find the sum of the arrays and print it.

- Solution:
  - Define 3 two-dimensional arrays and name them A, B, and C
  - Read values into A and B (use two for loops)
  - Find $C = A + B$. Two for loops are needed here too
#include<stdio.h>

void main()
{
    int A[5][5], B[5][5], C[5][5];
    int row, col;

    /* Read the first array */
    for( row = 0 ; row < 5 ; row ++ )
        for( col = 0; col < 5; col ++ )
            scanf("%d", &A[row][col] );

    /* Read the second array */
    for( row = 0 ; row < 5 ; row ++ )
        for( col = 0; col < 5; col ++ )
            scanf("%d", &B[row][col] );

    /* Find C = A + B */
    for( row = 0 ; row < 5 ; row ++ )
        for( col = 0; col < 5; col ++ )

    /* Print the sum array (C) */
    for( row = 0 ; row < 5 ; row ++ )
    {
        for( col = 0; col < 5; col ++ )
            printf("%d ", C[row][col] );
        printf("\n");
    }
}
Character Arrays or Strings

- Strings are groups of characters.
- Name of a student for example, is a string.
- Strings are defined using character arrays.
  
  Example: `char StudentName[30];`

- If we need to store many strings (the names of all students in a class for example), we use two-dimensional arrays of characters.
Two Dimensional Character Arrays

- A ‘\0’ character shows end of a string. The remaining variables in the array are not used.

<table>
<thead>
<tr>
<th>H</th>
<th>A</th>
<th>S</th>
<th>A</th>
<th>N</th>
<th>\0</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>E</td>
<td>H</td>
<td>M</td>
<td>E</td>
<td>T</td>
</tr>
<tr>
<td>H</td>
<td>A</td>
<td>T</td>
<td>I</td>
<td>C</td>
<td>E</td>
</tr>
<tr>
<td>B</td>
<td>O</td>
<td>R</td>
<td>A</td>
<td>\0</td>
<td></td>
</tr>
</tbody>
</table>
Example 4

- Write a program to define a two-dimensional character array to store the names of the students in a class (assume there are 10 students in the class).
- Read names into the array.
- Print the list of the students
#include<stdio.h>

void main()
{
    char name[10][30];
    int count;

    for( count = 0 ; count < 10; count ++ )
    {
        printf("Enter  student name:\n");
        scanf("%s",name[count]);
    }

    printf("The list of the students in the class is:\n");
    for( count = 0 ; count < 10; count ++ )
    {
        printf("%d - Student name is %s:\n", count, name[count]);
    }
}
Example 5

- Write a program to read the name and surname of 10 students. For each student, the program reads his/her final grade. The program prints the name and surname and the grade of each student and the difference between his/her grade and the average grade of the class.

- Example:

  If the average of the class is 55.0 then each line of the output will look like below:

  Hasan Demir grade = 48.0, difference with average : -7.0
#include<stdio.h>

void main()
{
    float grade[10], sum, average;
    char name[10][30], surname[10][30];
    int count;
    sum = 0;
    for( count = 0 ; count < 10; count ++ )
    {
        printf("Enter student name, surname and grade:\n");
        scanf("%s%s",name[count], surname[count]);
        scanf("%f", &grade[count]);
        sum = sum + grade[count];
    }
    average = sum /10;
    for(count = 0; count < 10; count ++ )
    {
        printf("%s %s grade = %4.1f, difference with average:%4.1f\n", name[count], surname[count], grade[count], grade[count]-average );
    }
}
Arrays are groups of variables with a common name.

Two-dimensional Arrays have rows and columns of elements having the same data types.

Two-dimensional array elements are accessed by row and column indexes.

Loops can be used to access elements of arrays.

Two-dimensional arrays of characters are used to store multiple strings.
Questions?