

Class 21 + 22

Arrays as Function Parameters;
Bubble Sort

1D arrays v 2D arrays

	1D example	2D example
Declaration	<code>int nums[5];</code>	<code>int grades[2][3];</code>
Initialization	<code>int nums[5] = {1, 2, 3, 4, 5};</code>	<code>int grades[2][3] = {{1, 2, 3}, {4, 5, 6}};</code>
Assign value to specific element	<code>nums[2] = 15; // stores 15 in nums at index 2</code>	<code>grades[0][2] = 10; // stores 10 at row index 0, column index 2</code>
Reference a specific element	<code>cout << nums[1];</code>	<code>cout << grades[1][1];</code>
Process array	Single for loop with counter initialized to 0	Nested for loops, with row and column counters initialized to 0

Accessing elements in 2D array

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

```
// read in values from user
int arr[3][4];
for(int row = 0; row < 3; row++){
    for(int col = 0; col < 4; col++){
        cin >> arr[row][col];
    }
}
```

2D arrays summarized in 1D arrays


```
int a[4][4];
// initialize a to random numbers
for(int row = 0; row < 4; row++){
    for(int col = 0; col < 4; col++){
        a[row][col] = rand()%10 + 1;
    }
}
int b[4] = {0};
// compute sum of each row of a and store in b
for(int row = 0; row < 4; row++){
    for(int col = 0; col < 4; col++){
        b[row] += a[row][col];
    }
}
```

Example 0

Goal: Fill a 5 x 5 array with random two-digit integers. Print the array. Count the number of odd numbers in each row. Print count of odd numbers in each row, and print all odd numbers in each row.

Plan:

- Declare 5 x 5 integer array
- Declare 1D int array with capacity 5
- Initialize all elements of 1D array to 0
- Using nested loops, fill 5 x 5 array with random numbers.
- Using nested loops, print the array in table format (rows and columns)
- Using nested loops, count the number of odd integers in row r and store the count in 1D array at index r
- Using nested loops, print the odd numbers in each row to the monitor

Arrays as function parameters

- Arrays can be used as input data for functions
- To call a function with an array as an argument:
 - Supply array name (or 2D array name) as input
 - Important: **Input array will be treated as call by reference.**
 - Any changes made to the array in the function are permanent

- Examples:

```
int main() {  
    int x[8] = {3,1,4,1,5,9,2,6}; // has average 3.875  
    peaks(x, 8); // prints 4 5 9 6  
    return 0;  
}
```

```
int main() {  
    int x[3][3] = {{3,1,4},{1,5,9}, {2,6,5}};  
    cout << maxIndex(x, 3, 3) << endl; // prints 1  
                                         // because the entry 9 is in row 1  
    return 0;  
}
```

Arrays as function parameters

- Set up an array parameter in the title line
 - For 1D, give an array declaration, **leaving capacity empty**
 - For 2D array, **leave row count empty but supply column count**
 - Will also need to include int parameters for capacity:
 - Count of rows
 - Count of columns

- Example:

```
void maxIndex(int first[][4], int second[], int r, int c) {  
    for (int i = 0; i < c; i++) {  
        second[i] = 0;  
        for (int j = 0; j < r; j++)  
            if (first[j][i] > first[second[i]][i])  
                second[i] = j;  
    }  
}
```

Example 1

- Write a function that fills a 1D array with random integers between 1 and 100
- Plan:
 - Declare int array inside main
 - Pass array to function
 - What is the title line for the function?
 - Inside function, populate array with random ints between 1 and 100
- How would you change the code to work with a 2D array?

Example 2

```
int main(){  
    int data[5] = {3, 1, 4, 1, 5};  
    int ans = maxElement(data, 5);  
    cout << ans << endl;  
    return 0;  
}
```

Code for maxElement function needs:

1. Title line
2. Code block

How would the code change to process a 2D array?

Example 3

- Write a function that checks if all integers in a 1D array are odd
- Plan:
 - Declare int array inside main
 - Pass array to function from example 1 to populate with ints
 - What is the title line for the function we need to write?
 - Inside function, how do we process array?
 - Return false if even number is encountered, otherwise return true

Example 4

- Write a function that returns the sum of a specific column of a 2D array of ints
- Plan:
 - Declare int array inside main
 - Pass array to function from example 1 to populate with ints
 - What is the title line for the function we need to write? What parameters do we need?
 - Inside function, how do we process array? How do we process a specific column?

Example 5

- mySwap + addNum
- Understand when we need to add & for the reference parameter

Example 6

- Write a **recursive** function called total to add up all the elements in an array

Bubble sort

<https://www.geeksforgeeks.org/bubble-sort/>