

# 2D Arrays Worksheet

1. Define a two-dimensional array named `temp` of three rows and four columns of type `int` such that the first row is initialized to 6, 8, 12, 9; the second row is initialized to 17, 5, 10, 6; and the third row is initialized to 14, 13, 16, 20.

```
int temp[3][4] = {  
    {6, 8, 12, 9},  
    {17, 5, 10, 6},  
    {14, 13, 16, 20}  
};
```

2. Suppose that array `temp` is as defined above. Write C++ statements to accomplish the following:

a. Output the content of the first row and first column element of `temp`.

```
cout << temp[0][0];
```

b. Output the content of the first row and last column element of `temp`.

```
cout << temp[0][3];
```

c. Output the content of the last row and first column element of `temp`.

```
cout << temp[2][0];
```

d. Output the content of the last row and last column element of `temp`.

```
cout << temp[2][3];
```

e. Output the sum of the last row.

```
int sum = 0;  
for (int i = 0; i < 4; ++i) {  
    sum += temp[2][i];  
}  
cout << sum;
```

f. Output the sum of the first column.

```
int sum = 0;  
for (int i = 0; i < 3; ++i) {  
    sum += temp[i][0];  
}
```

```
cout << sum;
```

g. Output the sum of the first two rows.

```
int sum = 0;
for (int i = 0; i < 4; ++i) {
    sum += temp[0][i];
    sum += temp[1][i];
}
cout << sum;
```

h. Output the sum of the last two rows.

```
int sum = 0;
for (int i = 0; i < 4; ++i) {
    sum += temp[1][i];
    sum += temp[2][i];
}
cout << sum;
```

3. Consider the following declaration:

```
int beta[3][3];
```

What is stored in beta after each of the following statements executes?

a. for (i = 0; i < 3; i++)  
 for (j = 0; j < 3; j++)  
 beta[i][j] = 0;

```
{
    {0, 0, 0}
    {0, 0, 0}
    {0, 0, 0}
}
```

b. for (i = 0; i < 3; i++)  
 for (j = 0; j < 3; j++)  
 beta[i][j] = i + j;

```
{
    {0, 1, 2}
    {1, 2, 3}
    {2, 3, 4}
}
```

```

c. for (i = 0; i < 3; i++)
    for (j = 0; j < 3; j++)
        beta[i][j] = i * j;
{
    {0, 0, 0}
    {0, 1, 2}
    {0, 2, 4}
}

```

```

d. for (i = 0; i < 3; i++)
    for (j = 0; j < 3; j++)
        beta[i][j] = 2 * (i + j) % 4

{
    {0, 2, 0}
    {2, 0, 2}
    {0, 2, 0}
}

```

4. Given: `int values[4][5]`  
Write a nested loop to set values as follows:

	[0]	[1]	[2]	[3]	[4]
[0]	1	2	3	4	5
[1]	1	2	3	4	5
[2]	1	2	3	4	5
[3]	1	2	3	4	5

```

for (int i = 0; i < 4; ++i) {
    for (int j = 0; j < 5; ++j) {
        values[i][j] = j + 1;
    }
}

```

5. Given: `int values[4][5]`  
Write a nested loop to set values as follows:

	[0]	[1]	[2]	[3]	[4]
[0]	0	1	2	3	4
[1]	1	2	3	4	5
[2]	2	3	4	5	6
[3]	3	4	5	6	7

```

for (int i = 0; i < 4; ++i) {
    for (int j = 0; j < 5; ++j) {
        values[i][j] = i + j;
    }
}

```

6. Given: `int matrix[5][5]`  
Write a nested loop to set matrix as follows:

```
    [0] [1] [2] [3] [4]
[0] 1   0   0   0   0
[1] 0   1   0   0   0
[2] 0   0   1   0   0
[3] 0   0   0   1   0
[4] 0   0   0   0   1
```

```
for (int i = 0; i < 5; ++i) {
    for (int j = 0; j < 5; ++j) {
        matrix[i][j] = (i == j);
    }
}
```

Given the following array of 5 rows and 5 columns, which contains the distances between cities:

		<b>[0]</b>	<b>[1]</b>	<b>[2]</b>	<b>[3]</b>	<b>[4]</b>
		<b>Albany</b>	<b>Boston</b>	<b>Hartford</b>	<b>NY</b>	<b>Phila</b>
<b>[0]</b>	<b>Albany</b>	0	171	115	141	240
<b>[1]</b>	<b>Boston</b>	171	0	103	194	333
<b>[2]</b>	<b>Hartford</b>	115	103	0	120	235
<b>[3]</b>	<b>NY</b>	141	194	120	0	104
<b>[4]</b>	<b>Phila</b>	240	333	235	104	0

Note that the city names are not in the array; the array contains the numeric entries only, which give the distance between the two cities represented by the row and column.

7. Write the statements to initialize an array distance with the mileage data given above.

```
int distance[5][5] = {
    { 0, 171, 115, 141, 240 },
    { 171, 0, 103, 194, 333 },
    { 115, 103, 0, 120, 235 },
    { 141, 194, 120, 0, 104 },
    { 240, 333, 235, 104, 0 },
};
```

8. Write a function which will print the following menu, read in two city numbers, and return the two city numbers entered:

To determine the mileage between cities, enter the numbers of two cities from the following list:

1: Albany	4: NY
2: Boston	5: Phila
3: Hartford	

Enter your city numbers:

```
void get_distances(int &city1, int &city2)
{
    cout << "To determine the mileage between cities, enter the numbers "
           "of two cities from the following list:\n\n";

    cout << "1: Albany\t4: NY\n";
    cout << "2: Boston\t5: Phila\n";
    cout << "3: Hartford\n\n";

    cout << "Enter your city numbers: ";
    cin >> city1 >> city2;
}
```

9. Write the statements to call your menu function and then print the distance between the two cities requested.

```
int city1, city2;

get_distances(city1, city2);

cout << "The distance between the cities is " << distance[city1][city2]
     << endl;
```