To:
Tenouk

C LAB WORKSHEET 10a_1
More On 2D Array Manipulation Part 3

1. More practice on 2D array data type.
2. Characters, strings and 2D array
3. Manipulating 2D array elements.
4. Tutorial references that should be used together with this worksheet are C & C++ array part 1 and C & C++ array part 2.

14. For the following question, use array bus[ ][ ] in the code. By referring to the given Figure, the rows represent the bus routes and the columns represent the days that the buses run. The numbers in the slots of the array show the number of passengers that were on a given route on a given day.

```
int bus[4][5];
```

Routes	Mon 0	Tue 1	Wed 2	Thu 3	Fri 4
0	8	12	9	7	10
1	5	7	3	0	4
2	20	15	18	21	14
3	6	9	5	8	11

a. Read the data into the array by receiving the data by columns. A portion of the output sample is given below.

```
----------Output----------
Give the number of passengers for day 1: 8 5 20 6
Give the number of passengers for day 2: 12 7 15 9
...
...
```

b. Print the number of passengers as shown in the Figure.
c. Print the total passengers for Mon.
d. Print the total number of passengers for the route stored in row 0.
e. Print the maximum number of passengers for the route stored in row 3.
f. Print the minimum number of passengers for the Thu.
g. Print the average number of passengers for all days and all routes.
h. Find the row with the largest number of passengers and print that number and the index.

The following are sample answers for g and h.

```
#include <stdio.h>

void main()
{
   int route, day = 0;
   float sum=0.0;
   int bus[4][5];

   for(day = 0; day <= 4; day = day + 1)
   {
      printf("Give the number of passengers for day %d: ", day+1);
      for(route = 0; route <= 3; route = route + 1)
         // scanf("%d", &bus[route][day]);
         scanf_s("%d", &bus[route][day], sizeof(bus));
   }
   printf("\n");
   // sum up all the element and find average
   for(day = 0; day <= 4; day = day + 1)
      for(route = 0; route <= 3; route = route + 1)
          sum = sum + bus[route][day];
   printf("The the average number of passengers for\n all days and all
routes is: %.2f\n", sum/20);
}
```

```
Give the number of passengers for day 1: 8 5 20 6
Give the number of passengers for day 2: 12 7 15 9
Give the number of passengers for day 3: 9 3 18 5
Give the number of passengers for day 4: 7 0 21 8
Give the number of passengers for day 5: 10 4 14 11

The the average number of passengers for
 all days and all routes is: 9.60
Press any key to continue . . .
```

```
#include <stdio.h>

void main()
{
```

The following are sample answers for a, b, c, d, e and f

```
#include <stdio.h>

void main()
{
   int route, day = 0;
   int bus[4][5];

   for(day = 0; day <= 4; day = day + 1)
   {
      printf("Give the number of passengers for day %d: ", day+1);
      for(route = 0; route <= 3; route = route + 1)
         // scanf("%d", &bus[route][day]);
         scanf_s("%d", &bus[route][day], sizeof(bus));
   }
   printf("\n");
   for(route = 0; route <= 3; route = route + 1)
   {
      for(day = 0; day <= 4; day = day + 1)
         printf("[%d][%d]=%2d ", route, day, bus[route][day]);
      printf("\n");
   }
}
```

```
Give the number of passengers for day 1: 8 5 20 6
Give the number of passengers for day 2: 12 7 15 9
Give the number of passengers for day 3: 9 3 18 5
Give the number of passengers for day 4: 7 0 21 8
Give the number of passengers for day 5: 10 4 14 11

[0][0]= 8  [0][1]=12  [0][2]= 9  [0][3]= 7  [0][4]=10
[1][0]= 5  [1][1]= 7  [1][2]= 3  [1][3]= 0  [1][4]= 4
[2][0]=20  [2][1]=15  [2][2]=18  [2][3]=21  [2][4]=14
[3][0]= 6  [3][1]= 9  [3][2]= 5  [3][3]= 8  [3][4]=11
Press any key to continue . . .
```

```
#include <stdio.h>

void main()
{
   int route, day = 0, sum=0;
   int bus[4][5];

   for(day = 0; day <= 4; day = day + 1)
   {
      printf("Give the number of passengers for day %d: ", day+1);
      for(route = 0; route <= 3; route = route + 1)
         // scanf("%d", &bus[route][day]);
         scanf_s("%d", &bus[route][day], sizeof(bus));
   }
   printf("\n");
   // only the route change, day is fix
   for(route = 0; route <= 3; route = route + 1)
   sum=sum + bus[route][0];
   printf("Total passenger on Monday is: %d\n", sum);
}
```

```
Give the number of passengers for day 1: 8 5 20 6
Give the number of passengers for day 2: 12 7 15 9
Give the number of passengers for day 3: 9 3 18 5
Give the number of passengers for day 4: 7 0 21 8
Give the number of passengers for day 5: 10 4 14 11

Total passenger on Monday is: 39
Press any key to continue . . .
```

```
#include <stdio.h>

void main()
{
   int route, day = 0, sum=0;
   int bus[4][5];

   for(day = 0; day <= 4; day = day + 1)
   {
      printf("Give the number of passengers for day %d: ", day+1);
      for(route = 0; route <= 3; route = route + 1)
         // scanf("%d", &bus[route][day]);
         scanf_s("%d", &bus[route][day], sizeof(bus));
   }
   printf("\n");
   // only the day change, route is fix
   for(day = 0; day <= 4; day = day + 1)
      sum=sum + bus[0][day];
```

```
    int route, day = 0, largest, daymax, routemax;
    int bus[4][5];

    for(day = 0; day <= 4; day = day + 1)
    {
       printf("Give the number of passengers for day %d: ", day+1);
       for(route = 0; route <= 3; route = route + 1)
             // scanf("%d", &bus[route][day]);
             scanf_s("%d", &bus[route][day], sizeof(bus));
    }
    printf("\n");
    // find the largest and its index
    largest = bus[0][0];
    for(day = 1; day <= 4; day = day + 1)
       for(route = 1; route <= 3; route = route + 1)
       {
          if(bus[day][route] > largest)
          {
             largest = bus[day][route];
             daymax = day;
             routemax = route;
          }
       }
    printf("The largest is %d with index [%d][%d] or route %d\n", largest,
daymax, routemax, daymax);
}
```

```
Give the number of passengers for day 1: 8 5 20 6
Give the number of passengers for day 2: 12 7 15 9
Give the number of passengers for day 3: 9 3 18 5
Give the number of passengers for day 4: 7 0 21 8
Give the number of passengers for day 5: 10 4 14 11

The largest is 21 with index [2][3] or route 2
Press any key to continue . . .
```

```
    printf("Total passenger for route 0 is: %d\n", sum);
}
```

```
Give the number of passengers for day 1: 8 5 20 6
Give the number of passengers for day 2: 12 7 15 9
Give the number of passengers for day 3: 9 3 18 5
Give the number of passengers for day 4: 7 0 21 8
Give the number of passengers for day 5: 10 4 14 11

Total passenger for route 0 is: 46
Press any key to continue . . .
```

```
#include <stdio.h>

void main()
{
    int route, day = 0, largest;
    int bus[4][5];

    for(day = 0; day <= 4; day = day + 1)
    {
       printf("Give the number of passengers for day %d: ", day+1);
       for(route = 0; route <= 3; route = route + 1)
          // scanf("%d", &bus[route][day]);
          scanf_s("%d", &bus[route][day], sizeof(bus));
    }
    printf("\n");
    // only the day change, route is fix
    largest = bus[3][0];
    for(day = 1; day <= 4; day = day + 1)
    {
       if(bus[3][day] > largest)
       largest = bus[3][day];
    }
    printf("maximum number of passengers for the route 3 is: %d\n", largest);
}
```

```
Give the number of passengers for day 1: 8 5 20 6
Give the number of passengers for day 2: 12 7 15 9
Give the number of passengers for day 3: 9 3 18 5
Give the number of passengers for day 4: 7 0 21 8
Give the number of passengers for day 5: 10 4 14 11

maximum number of passengers for the route 3 is: 11
Press any key to continue . . .
```

```
#include <stdio.h>

void main()
{
    int route, day = 0, smallest;
    int bus[4][5];

    for(day = 0; day <= 4; day = day + 1)
    {
       printf("Give the number of passengers for day %d: ", day+1);
       for(route = 0; route <= 3; route = route + 1)
          // scanf("%d", &bus[route][day]);
          scanf_s("%d", &bus[route][day], sizeof(bus));
    }
    printf("\n");
    // only the route change, day is fix
    smallest = bus[0][3];
    for(route = 0; route <= 3; route = route + 1)
    {
       if(bus[route][3] < smallest)
          smallest = bus[route][3];
    }
    printf("The minimum number of passengers for the Thu is: %d\n", smallest);
}
```

```
Give the number of passengers for day 1: 8 5 20 6
Give the number of passengers for day 2: 12 7 15 9
Give the number of passengers for day 3: 9 3 18 5
Give the number of passengers for day 4: 7 0 21 8
Give the number of passengers for day 5: 10 4 14 11

The minimum number of passengers for the Thu is: 0
Press any key to continue . . .
```

15. In the following code, how many of the printf()'s are part of the loop?

```
#include <stdio.h>

void main()
{
    int j;
    for(j = 0; j <= 4; j = j + 1)
        printf("(4 - %d) = %d\t", j, 4 - j);
    printf("\n");
}
```

Just the first printf() is part of the loop because there is no curly braces used for the for loop body, then only the first statement immediately after the for loop statement is part of the for loop body.

16. Now let us place the j loop inside an i loop, making the j loop nested. We want the j loop and the last printf() to be inside the i loop, so a pair of braces are needed. When i is 0, we start the j loop. Here, j start at 0 and goes to 4, each time printing the value of i and j. For the first line, 0 0 0 1 0 2 0 3 0 4 will be printed because i stays fixed at 0. Show the output.

```
#include <stdio.h>

void main()
{
    int i, j;
    for(i = 0; i <= 2; i = i + 1)
    {
        for(j = 0; j <= 4; j = j + 1)
            printf("i j = %d %d  ", i, j);
        printf("\n");
    }
}
```

```
i j = 0 0  i j = 0 1  i j = 0 2  i j = 0 3  i j = 0 4
i j = 1 0  i j = 1 1  i j = 1 2  i j = 1 3  i j = 1 4
i j = 2 0  i j = 2 1  i j = 2 2  i j = 2 3  i j = 2 4
Press any key to continue . . .
```

17. Show the output when i and j are reversed in the printf().

```
#include <stdio.h>

void main()
{
    int i, j;
    for(i = 0; i <= 2; i = i + 1)
    {
        for(j = 0; j <= 4; j = j + 1)
            printf("j i = %d %d  ", j, i);
        printf("\n");
    }
}
```

```
j i = 0 0 j i = 1 0 j i = 2 0 j i = 3 0 j i = 4 0
j i = 0 1 j i = 1 1 j i = 2 1 j i = 3 1 j i = 4 1
j i = 0 2 j i = 1 2 j i = 2 2 j i = 3 2 j i = 4 2
Press any key to continue . . .
```

18. Rewrite only the for loop statements so that the output is as shown below.

```
0 0  1 0
0 1  1 1
0 2  1 2
0 3  1 3
```

```
#include <stdio.h>

void main()
{
    int i, j;
    for(i = 0; i <= 3; i = i + 1)
    {
        for(j = 0; j <= 1; j = j + 1)
            printf("%d %d ", j, i);
        printf("\n");
    }
}
```

```
0 0   1 0
0 1   1 1
0 2   1 2
0 3   1 3
Press any key to continue . . .
```

19. Now instead of printing both i and j, let us print their sum and remove the braces. Show the output.

```
#include <stdio.h>

void main()
{
    int i, j;
    for(i = 0; i <= 3; i = i + 1)
        for(j = 0; j <= 1; j = j + 1)
            printf("(j+i)=%d\n", j + i);
        printf("\n");
}
```

```
(j+i)=0
(j+i)=1
(j+i)=1
(j+i)=2
(j+i)=2
(j+i)=3
(j+i)=3
(j+i)=4
Press any key to continue . . .
```

20. In the following Figure we have a 2D array called A[3][5]. The first index is called the row and the second is the column. A[1][3] refers to the slot where the row is 1 and the column is 3. The number here is 90. A[ ][ ] is defined as:

```
int A[3][5];
```

Take note that in the declaration the index of the array represent the size. Here we have 3 x 5 = 15 slots. However in a program the number in the square brackets are array's indexes.

	0	1	2	3	4
0	50	120	30	150	80
1	140	70	110	90	100
2	30	20	0	60	40

Column indexes

Row indexes

a. What is A[0][1] and A[2][4]?
b. If i is 1 and j is 2, what is A[i][j] and A[j][i]?

a. A[0][1] = 120 and A[2][4] = 40.
b. A[i][j]=A[1][2]=110 and A[j][i]=A[2][1]=20.

21. In the following code, when i is 0, j varies from 0 to 4. While i is 0, the output would be 50, 120, 30, 150, 80. Complete the code snippet and show the complete output.

```
for(i = 0; i <= 2; i = i + 1)
        {
          for(j = 0; j <= 4; j = j + 1)
              printf("A[%d][%d] = %d", i, j, A[i][j]);
          printf("\n");
        }
```

```
#include <stdio.h>

void main()
{
   int i, j, A[3][5] = {{50, 120, 30, 150, 80}};
   for(i = 0; i <= 2; i = i + 1)
   {
      for(j = 0; j <= 4; j = j + 1)
         printf("A[%d][%d]=%3d ", i, j, A[i][j]);
      printf("\n");
   }
}
```

```
A[0][0]= 50  A[0][1]=120  A[0][2]= 30  A[0][3]=150  A[0][4]= 80
A[1][0]=  0  A[1][1]=  0  A[1][2]=  0  A[1][3]=  0  A[1][4]=  0
A[2][0]=  0  A[2][1]=  0  A[2][2]=  0  A[2][3]=  0  A[2][4]=  0
Press any key to continue . . .
```

22. Now let us swap i and j subscripts in the printf(). Here, j would be the row subscript and j shouldn't go up to 4 because we don't have that many rows. Hence, the for loop statements have to be rewritten as shown. Show the complete code and the output.

```
for(i = 0; i <= 4; i = i + 1)
        {
          for(j = 0; j <= 2; j = j + 1)
             printf("A[%d][%d] = %d", j, i, A[j][i]);
          printf("\n");
        }
```

```
#include <stdio.h>

void main()
{
   int i, j, A[3][5] = {{50, 120, 30, 150, 80}};
   for(i = 0; i <= 4; i = i + 1)
   {
      for(j = 0; j <= 2; j = j + 1)
         printf("A[%d][%d]=%3d ", j, i, A[j][i]);
      printf("\n");
   }
}
```

```
A[0][0]= 50  A[1][0]=  0  A[2][0]=  0
A[0][1]=120  A[1][1]=  0  A[2][1]=  0
A[0][2]= 30  A[1][2]=  0  A[2][2]=  0
A[0][3]=150  A[1][3]=  0  A[2][3]=  0
A[0][4]= 80  A[1][4]=  0  A[2][4]=  0
Press any key to continue . . .
```

23. The following code goes through the entire array. At each slot it checks if the value is greater than 100.

```
#include <stdio.h>

void main()
{
    int i, j, count = 0;
    int A[3][5] = {50, 120, 30, 150, 80, 140, 70, 110, 90, 100, 30, 20, 0, 60,
40};

    for(i = 0; i <= 4; i = i + 1)
        for(j = 0; j <= 2; j = j + 1)
            if(A[j][i] > 100)
                count = count + 1;
            printf("count = %d\n", count);
}
```

a. What will be printed?
b. Are the slots of the array visited row-wise or column-wise?
c. Why aren't there any braces in the code?

```
count = 4
Press any key to continue . . .
```

a. The number of array element that greater than 100 will be printed.
b. Column-wise because of the A[j][i] used in the if statement.
c. Because the if statement only have one statement that is count = count + 1; and this if statement is the only one of the second for statement. Hence no need to use curly braces for the second for and the if statements.

24. Write a complete code to find the largest number in each column. Here is a start.

```
    ...
    ...
 int largest;
 for(i = 0; i <= 4; i = i + 1)
 {
     largest = A[0][i];
     for(j = 1; j <= 2; j = j + 1)
         ...
         ...
```

```
#include <stdio.h>

void main()
{
   int i, j, A[3][5] = {{50, 120, 30, 150, 80}, {140, 70, 110, 90, 100}, {30, 20, 0, 60, 40}};

   int largest;

   // print the whole array
   for(j = 0; j <= 2; j = j + 1)
   {
      for(i = 0; i <= 4; i = i + 1)
         printf("%3d ", A[j][i]);
      printf("\n");
    }
   printf("\n");
   for(i = 0; i <= 4; i = i + 1)
   {
      largest = A[0][i];
      for(j = 1; j <= 2; j = j + 1)
         if(A[j][i] > largest)
            largest = A[j][i];
      printf("The largest in column %d is: %d\n", i, largest);
   }
}
```

```
 50 120  30 150  80
140  70 110  90 100
 30  20   0  60  40

The largest in column 0 is: 140
The largest in column 1 is: 120
The largest in column 2 is: 110
The largest in column 3 is: 150
The largest in column 4 is: 100
Press any key to continue . . .
```

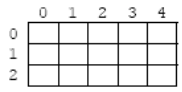
25. Next, modify the code to find the smallest number in each column.

```
#include <stdio.h>

void main()
{
   int i, j, A[3][5] = {{50, 120, 30, 150, 80}, {140, 70, 110, 90, 100}, {30, 20, 0, 60, 40}};
   int smallest;

   // print the whole array
   for(j = 0; j <= 2; j = j + 1)
   {
      for(i = 0; i <= 4; i = i + 1)
         printf("%3d ", A[j][i]);
      printf("\n");
   }
   printf("\n");
   for(i = 0; i <= 4; i = i + 1)
   {
      smallest = A[0][i];
      for(j = 1; j <= 2; j = j + 1)
         if(A[j][i] < smallest)
      smallest = A[j][i];
      printf("The smallest in column %d is: %d\n", i, smallest);
   }
}
```

```
 50 120  30 150  80
140  70 110  90 100
 30  20   0  60  40

The smallest in column 0 is: 30
The smallest in column 1 is: 20
The smallest in column 2 is: 0
The smallest in column 3 is: 60
The smallest in column 4 is: 40
Press any key to continue . . .
```

26. Using the array of the following code snippet, read "I", "do" and "well" into the character array. Notice that for 2D array, only one subscript is specified. One subscript works because we are reading in strings. Complete the code and show the contents of the array, including the null character.

```
    ...
    ...
char A[3][5];
for(i = 0; i <= 2; i = i + 1)
    scanf_s("%s", &A[i]);
    ...
    ...
```




```
#include <stdio.h>

void main()
{
   int i,j;
   char A[3][5];

   for(i = 0; i <= 2; i = i + 1)
      // scanf("%s", &A[i]);
      scanf_s("%s", &A[i], sizeof(A[3]));
   for(i = 0; i <= 2; i = i + 1)
   {
      for(j = 0; j <= 4; j++)
         printf("A[%d][%d]->%c ", i, j, A[i][j]);
      printf("\n");
   }
}
```

```
I do well
A[0][0]->I  A[0][1]->   A[0][2]->²  A[0][3]->²  A[0][4]->²
A[1][0]->d  A[1][1]->o  A[1][2]->   A[1][3]->²  A[1][4]->²
A[2][0]->w  A[2][1]->e  A[2][2]->l  A[2][3]->l  A[2][4]->
Press any key to continue . . .
```

	0	1	2	3	4
0	I	\0			
1	d	o	\0		
2	w	e	l	l	\0

The shaded slots filled with garbage.

27. Write a single loop to print the array of strings in the previous question.

```
#include <stdio.h>

void main()
{
   int i;
   char A[3][5];

   for(i = 0; i <= 2; i = i + 1)
      // scanf("%s", &A[i]);
      scanf_s("%s", &A[i], sizeof(A[3]));
   for(i = 0; i <= 2; i = i + 1)
      printf("%s ", A[i]);
   printf("\n");
}
```

```
I do well
I do well
Press any key to continue . . .
```