

To:
Tenouk

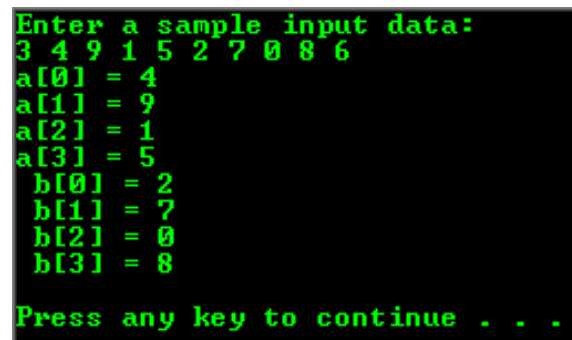
C LAB WORKSHEET 9a

More On C & C++ Array Manipulation Part 3

1. More array manipulations.
2. Tutorial references that should be used together with this worksheet are [array part 1](#) and [array part 2](#).
3. Show the content of the arrays for the following a[] and b[], using the given input data and study the code and the output. Enter the data line by line. Not all data items may be used.
Sample input: 3 4 9 1 5 2 7 0 8 6

```
#include <stdio.h>
```

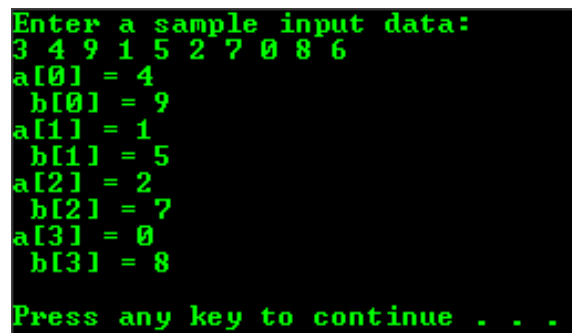
```
void main()
{
    int i, a[12], b[12], k;
    printf("Enter a sample input data: \n");
    scanf_s("%d", &k);
    for(i = 0; i <= k; i = i + 1)
    {
        scanf_s("%d", &a[i]);
        printf("a[%d] = %d\n", i, a[i]);
    }
    for(i = 0; i <= k; i = i + 1)
    {
        scanf_s("%d", &b[i]);
        printf(" b[%d] = %d\n", i, b[i]);
    }
    printf("\n");
}
```



```
Enter a sample input data:
3 4 9 1 5 2 7 0 8 6
a[0] = 4
a[1] = 9
a[2] = 1
a[3] = 5
 b[0] = 2
 b[1] = 7
 b[2] = 0
 b[3] = 8
Press any key to continue . . .
```

```
#include <stdio.h>
```

```
void main()
{
    int i, a[12], b[12], k;
    printf("Enter a sample input data: \n");
    scanf_s("%d", &k);
    for(i = 0; i <= k; i = i + 1)
    {
        scanf_s("%d", &a[i]);
        printf("a[%d] = %d\n", i, a[i]);
        scanf_s("%d", &b[i]);
        printf(" b[%d] = %d\n", i, b[i]);
    }
    printf("\n");
}
```

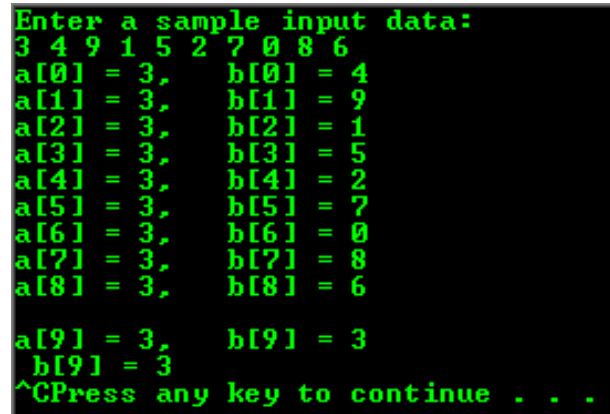


```
Enter a sample input data:
3 4 9 1 5 2 7 0 8 6
a[0] = 4
 b[0] = 9
a[1] = 1
 b[1] = 5
a[2] = 2
 b[2] = 7
a[3] = 0
 b[3] = 8
Press any key to continue . . .
```

To terminate the program you need to press Ctrl + C for PC/IBM compatible.

```
#include <stdio.h>
```

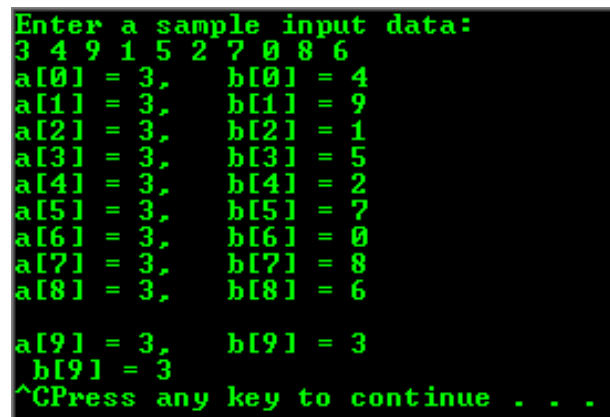
```
void main()
{
    int i, a[12], b[12], k;
    printf("Enter a sample input data: \n");
    scanf_s("%d", &k);
    for(i = 0; k != 1; i = i + 1)
    {
        a[i] = k;
        b[i] = k;
        scanf_s("%d", &b[i]);
        printf("a[%d] = %d ", i, a[i]);
        printf(" b[%d] = %d\n", i, b[i]);
    }
    printf("\n");
}
```



```
Enter a sample input data:
3 4 9 1 5 2 7 0 8 6
a[0] = 3,    b[0] = 4
a[1] = 3,    b[1] = 9
a[2] = 3,    b[2] = 1
a[3] = 3,    b[3] = 5
a[4] = 3,    b[4] = 2
a[5] = 3,    b[5] = 7
a[6] = 3,    b[6] = 0
a[7] = 3,    b[7] = 8
a[8] = 3,    b[8] = 6
a[9] = 3,    b[9] = 3
  b[9] = 3
^CPress any key to continue . . .
```

```
#include <stdio.h>
```

```
void main()
{
    int i, a[12], b[12], k;
    printf("Enter a sample input data: \n");
    scanf_s("%d", &k);
    for(i = 0; k != 8; i = i + 1)
    {
        a[i] = k;
        scanf_s("%d", &b[i]);
        scanf_s("%d", &k);
        printf("a[%d] = %d\n", i, a[i]);
        printf(" b[%d] = %d\n", i, b[i]);
    }
    printf("\n");
}
```



```
Enter a sample input data:
3 4 9 1 5 2 7 0 8 6
a[0] = 3,    b[0] = 4
a[1] = 3,    b[1] = 9
a[2] = 3,    b[2] = 1
a[3] = 3,    b[3] = 5
a[4] = 3,    b[4] = 2
a[5] = 3,    b[5] = 7
a[6] = 3,    b[6] = 0
a[7] = 3,    b[7] = 8
a[8] = 3,    b[8] = 6
a[9] = 3,    b[9] = 3
  b[9] = 3
^CPress any key to continue . . .
```

For the following program, you need to modify the code to see the a[] and b[].

```
#include <stdio.h>
```

```
void main()
{
    int i, a[12], b[12], k;
    printf("Enter a sample input data: \n");
    scanf_s("%d", &k);
    for(i = 0; k > 1; i = i + 1)
    {
        a[i] = k;
        scanf_s("%d", &k);
    }
    scanf_s("%d", &k);
```

```
#include <stdio.h>
```

```
void main()
{
    int i, a[12], b[12], k;
    printf("Enter a sample input data: \n");
    scanf_s("%d", &k);
    for(i = 0; k > 1; i = i + 1)
    {
        a[i] = k;
        scanf_s("%d", &k);
        printf("a[%d]= %d\n", i, a[i]);
    }
    scanf_s("%d", &k);
    for(i = 0; k > 1; i = i + 1)
    {
```

```

for(i = 0; k > 1; i = i + 1)
{
    b[i] = k;
    scanf_s("%d", &k);
}
printf("\n");
}

```

```

b[i] = k;
scanf_s("%d", &k);
printf(" b[%d]= %d\n", i, b[i]);
}
printf("\n");
}

```

```

Enter a sample input data:
3 4 9 1 5 2 7 0 8 6
a[0]= 3
a[1]= 4
a[2]= 9
b[0]= 5
b[1]= 2
b[2]= 7
Press any key to continue . . . _

```

4. Create a program that read 10 floats into an array and print their average.

The following is a pseudocode sample.

Declare float and other variables to hold data.

float data[10], sum; int i;

Prompt for 10 floats.

printf("Enter 10 floats:\n");

Starts the for loop.

for(i=0;i<=9;i++)

Read & store the data.

scanf_s("%f", &data[i], sizeof(float));

Sum up all the floats.

sum = sum + data

[i];

The for loop stops.

Calculate and print the sum's average.

average = (sum/i);

The program stops.

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
float data[10]={0.0}, sum=0.0;
```

```
int i;
```

```
printf("Enter 10 floats:\n");
```

```
for(i=0;i<=9;i++)
```

```
{
```

```
scanf_s("%f", &data[i], sizeof(float));
```

```
sum = sum + data[i];
```

```
}
```

```
printf(" i = %d, sum = %.2f\n", i, sum);
```

```
printf("The average is = %.2f\n", (sum/i));
```

```
return 0;
```

```
}
```

```

Enter 10 floats:
2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0 2.0
i = 10, sum = 20.00
The average is = 2.00
Press any key to continue . . . _

```

4. Initialize an array with 6 integers. Read another integer and add that number to each element in the array. Then print the array.

The following is a pseudocode sample.

Declare and initialize an array with 6 integers and other variable to hold data.

```
int num[6] = {3, 5, 7, 2, 12, 10},
num2, i;
```

Prompt for another integer.

```
printf("Enter an integer: ");
```

Read & store the integer.

```
scanf_s("%d", &num2,
```

```
sizeof(int));
```

Start a for loop to add the number to the each existing array element and print those array element.

```
printf("New array element
are: \n");
```

```
for(i=0;i<=5;i++)
```

```
{
```

```
num[i] = num[i] +
num2;
```

```
printf("%d ", num
[i]);
```

```
}
```

The loop stops.

The program stops.

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int num[6] = {3, 5, 7, 2, 12, 10}, num2, i;
```

```
printf("Enter an integer: ");
```

```
scanf_s("%d", &num2, sizeof(int));
```

```
printf("Original array element are: \n");
```

```
for(i=0;i<=5;i++)
```

```
printf("%d ", num[i]);
```

```
printf("\nNew array element are: \n");
```

```
for(i=0;i<=5;i++)
```

```
{
```

```
num[i] = num[i] + num2;
```

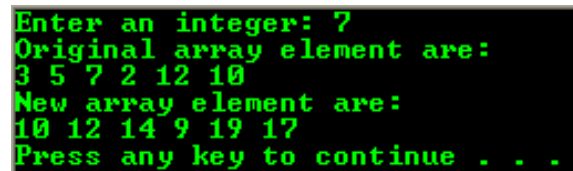
```
printf("%d ", num[i]);
```

```
}
```

```
printf("\n");
```

```
return 0;
```

```
}
```



```
Enter an integer: 7
Original array element are:
3 5 7 2 12 10
New array element are:
10 12 14 9 19 17
Press any key to continue . . .
```

4. Initialize an array with 6 integers. Read another integer and print the value of each array element that is less than this read-in integer and print the number of elements that were not printed. For example, if the array were 40, 80, 20, 50, 90, 30 and an 80 were read in, then the output would be something like:

```
-----Output-----
```

```
40 20 50 30
```

```
count = 2
```

The following is a pseudocode sample.

Declare and initialize an array with 6 integers and other

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
int num[6] = {40, 70, 90, 20, 50, 35}, num2, i,
count=0;
```

```
printf("Enter an integer: ");
```

```
scanf_s("%d", &num2, sizeof(int));
```

```
for(i=0;i<=5;i++)
```

```
{
```

```
if(num[i] < num2)
```

```
printf("%d ", num[i]);
```

```
if(num[i] > num2)
```

```
count = count + 1;
```

```
}
```

```
printf("\ncount = %d\n", count);
```

```
return 0;
```

variables to hold data.

```
int num[6] = {40, 70, 90,
20, 50, 35}, num2, i, count=0;
Prompt for an integer and read &
store the integer.
```

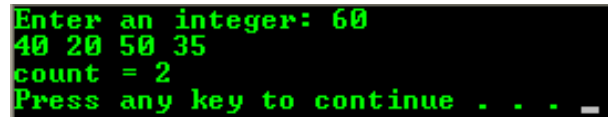
```
printf("Enter an integer:
\n");
scanf_s("%d", &num2,
sizeof(int));
Scan all the array element and
compare to num2. Print array
element that less than num2.
for(i=0;i<=5;i++)
{
    if(num[i] < num2)
        printf("%d
", num[i]);
    if(num[i] > num2)
        count =
count + 1;
}
```

The loop stops.

Print the number of the rest of
the array element.

```
printf("\ncount = %d\n",
count);
The program stops.
```

```
}
```



```
Enter an integer: 60
40 20 50 35
count = 2
Press any key to continue . . . _
```

7. Initialize two integer arrays, **a[6]** and **b[6]**. First print their elements in two rows, and then print them out in two columns.

The following is a pseudocode sample.

Declare and initialize two integer arrays and other variable to hold data.

```
int a[6]={15, 13, 27, 2, 8,
10}, b[6]={12, 10, 19, 7, 4, 20}, i;
Scan the array elements and
print in rows.
```

```
for(i=0;i<=5;i++)
    printf("%d ", a[i]);
for(i=0;i<=5;i++)
    printf("%d ", a[i]);
```

Scan the array elements and
print in columns.

```
for(i=0;i<=5;i++)
    printf("%d %d\n",
a[i], b[i]);
```

```
#include <stdio.h>
```

```
int main()
```

```
{
    int a[6]={15, 13, 27, 2, 8, 10}, b[6]={12, 10, 19, 7,
4, 20}, i;
```

```
printf("a[i] = ");
for(i=0;i<=5;i++)
    printf("%d ", a[i]);
printf("\nb[i] = ");
for(i=0;i<=5;i++)
    printf("%d ", a[i]);
printf("\n\na[i]\tb[i]\n");
for(i=0;i<=5;i++)
    printf("%d\t%d\n", a[i], b[i]);
return 0;
```

```
}
```

The program stops.

```
a[i] = 15 13 27 2 8 10
b[i] = 15 13 27 2 8 10

a[i]    b[i]
15      12
13      10
27      19
2        7
8        4
10      20
Press any key to continue . . .
```

7. Scan floating point numbers into an array until the sum of those numbers exceed 100. At the index where the sum becomes greater than 100, store a -1.0 to denote the end of the array. Now print the number of elements in the array, not including the -1.0.

The following is a pseudocode sample.

Declare variables to store data.

```
float num[20], sum=0.0;
int i;
```

Prompt floats from user.

```
printf("Enter floats: \n");
```

Read and store those floats in an array.

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

```
    scanf_s("%f",
&num[i], sizeof(float));
```

Sum up all the element until the sum greater than 100. Store -1.0 into the array where the sum > 100.

```
for(i=0;sum<100;i++)
```

```
{
    scanf_s("%f",
&num[i], sizeof(float));
    sum = sum + num
```

```
[i];
```

```
    if(sum > 100)
        num[i] = -1.0;
```

```
}
```

The loop stops.

Print the array elements.

```
for(i=0;num[i] != -1.0;i++)
    printf("%.2f ", num
```

```
[i]);
```

The loop stops.

The program stops.

```
#include <stdio.h>
```

```
int main()
```

```
{
    float num[20], sum=0.0;
    int i;
    printf("Enter floats: \n");
    for(i=0;sum<100;i++)
    {
        scanf_s("%f", &num[i], sizeof(float));
        sum = sum + num[i];
        if(sum > 100)
            num[i] = -1.0;
        printf("num[%d] = %.2f\n", i, num[i]);
    }
    for(i=0;num[i] != -1.0;i++)
        printf("%.2f ", num[i]);
    printf("\n");
    return 0;
}
```

```
Enter floats:
15.5
num[0] = 15.50
25.5
num[1] = 25.50
17.4
num[2] = 17.40
20.2
num[3] = 20.20
30.3
num[4] = -1.00
15.50 25.50 17.40 20.20
Press any key to continue . . .
```

7. Initialize an integer array with 6 elements. Find the lowest element and swap it with the number in the first index. For example, if the array is 40, 80, 20, 50, 90, 30 then after the swap the array should become 20, 80, 40, 50, 90, 30.

The following is a pseudocode sample.

Declare and initialize an integer array with 6 elements and other variables to store data.

```
int i, k[6]={40, 80, 20, 50, 90, 30},
smallest, item_num, swap;
Scan the array for the lowest
element and store it with the
index where the lowest occurred.
smallest = k[0];
```

```
    item_num = 1;
    for(i=0;i<=5;i++)
    {
        if(k[i] < smallest)
        {
            smallest = k[i];
            item_num = i;
            swap=k
[item_num];
        }
    }
```

The loop stop.

Do the swapping.

```
k[item_num]=k[0];
k[0]=swap;
```

Print the element.

```
for(i = 0; i <=5; i = i + 1)
    printf("%d ", k[i]);
```

The loop stop.

The program stop.

```
#include <stdio.h>
```

```
int main(void)
```

```
{
    int i, k[6]={40,80,20,50,90,30}, smallest,
item_num, swap;
    // assign element no 1 to variable smallest
    smallest = k[0];
    // initialize item_num to 1 n finding the lowest
element's index
    item_num = 1;
    // print the original array
    printf("Original array: ");
    for(i = 0; i <=5; i = i + 1)
        printf("%d ", k[i]);
    // start the for loop with index 1 coz 0 already
// assigned to smallest
    for(i = 1; i <=5; i = i + 1)
    {
        // if the element is < smallest...
        if(k[i] < smallest)
        {
            // assign the element to smallest
            smallest = k[i];
            // assign the count to item_num as the
lowest's index
            item_num = i;
            // assign the smallest temporarily for
swapping
            swap=k[item_num];
        }
    }
    // do the swapping, the first element assign to
the lowest element
    k[item_num]=k[0];
    // the lowest element assign to the first element
    k[0]=swap;
    // print the elements
    printf("\nNew array: ");
    for(i = 0; i <=5; i = i + 1)
        printf("%d ", k[i]);
    printf("\n");
    return 0;
}
```

```
Original array: 40 80 20 50 90 30
New array: 20 80 40 50 90 30
Press any key to continue . . .
```

