CS1050 – Lab 6
Fall 2021

Concepts to Practice

- Arrays
- Passing arrays to functions
- Expand the prelab

Submission Information

Submit this assignment by following the instructions given by your TA. SUBMIT ONLY the .c file (no a.out or executable file is required). All of the lab assignments must be submitted before the end of the lab.

Use the following submit command:

mucs submit <class> <assignment_> <filename>

For example:

mucs submit 1050 lab6 lab6.c

Description

For the lab assignment, you need to declare 3 arrays in your main() function. By passing these 3 arrays to functions that you write, you should be able to initialize two of them to the specified values, multiply their values together to form the 3rd array, and print all of these arrays.

Important note: **Global variables are not allowed**!

In your main() function, you can **ONLY** do the following things:

- Declare variables
- Call the printf() function
- Call any function that you write (including assigning a variable to the return value of your function if you wish)

Overall, you program should:

- Declare 3 integer arrays that have 100 elements each.
- Initialize an array to hold the values 1,4,7,…,31 (inclusive)
- Initialize a second array to hold the values 2,7,12,…,52 (inclusive)
- Multiply the first value of the first array with the first value of the second array, the 2nd value of the first array with the 2nd value of the second array, etc. and place the results in the corresponding elements of the output array.
- Print the two "input" arrays as well as the "output" array with corresponding elements on the same lines (as shown in the sample output)
- Do this procedure again with the 2 input arrays having values 22,20,18,16,…,0 and 84,77,70,63,…,0 respectively (note that you will call your function to multiply the two together as well, and will print everything out again too – see sample output).

Honors Extension

Write a second function that instead of multiplying the corresponding elements of each array, multiplies the first element of array 1 with the last element of array 2, the second element of array 1 with the second to the last element of array 2, and so on. Print your "reverse multiply" results after each "regular multiply" result (see sample output). I just wrote a 2nd function to do this, but if you are careful you could just add a parameter to your MultArrays() function instead (so it would serve both purposes!).

Functions

You may write any functions you wish to implement in this program. However, don't forget that you cannot do all of the work in main() – most of the work must be done in functions. As a hint to help you along, here is the prototype for the function I wrote to multiply the arrays together:

- void MultArrays(int array1[], int array2[], int arrayOut[], int size);

Hints

- Start with printing the input arrays separately as you did in your prelab. If you have time, you can come back later and write a new function that prints them with corresponding elements on the same line.
- Focus on making the "First Arrays" work first. Once that works, submit your work and make changes necessary to support the "Second Arrays".
- You can assume both input arrays have the same number of elements when you are multiplying their elements to form the output array. However, notice that the first 2 arrays have 11 elements, and the second 2 arrays have 12 elements. You may need to parameterize your functions to deal with this.

Sample Non-honors Output

```
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab6$ compile lab6.c
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab6$ ./a.out
First Arrays:
        Array 1 Element  0 =  1 Array 2 Element  0 =  2 Array 3 Element  0 =    2
        Array 1 Element  1 =  4 Array 2 Element  1 =  7 Array 3 Element  1 =   28
        Array 1 Element  2 =  7 Array 2 Element  2 = 12 Array 3 Element  2 =   84
        Array 1 Element  3 = 10 Array 2 Element  3 = 17 Array 3 Element  3 =  170
        Array 1 Element  4 = 13 Array 2 Element  4 = 22 Array 3 Element  4 =  286
        Array 1 Element  5 = 16 Array 2 Element  5 = 27 Array 3 Element  5 =  432
        Array 1 Element  6 = 19 Array 2 Element  6 = 32 Array 3 Element  6 =  608
        Array 1 Element  7 = 22 Array 2 Element  7 = 37 Array 3 Element  7 =  814
        Array 1 Element  8 = 25 Array 2 Element  8 = 42 Array 3 Element  8 = 1050
        Array 1 Element  9 = 28 Array 2 Element  9 = 47 Array 3 Element  9 = 1316
        Array 1 Element 10 = 31 Array 2 Element 10 = 52 Array 3 Element 10 = 1612

Second Arrays:
        Array 1 Element  0 = 22 Array 2 Element  0 = 84 Array 3 Element  0 = 1848
        Array 1 Element  1 = 20 Array 2 Element  1 = 77 Array 3 Element  1 = 1540
        Array 1 Element  2 = 18 Array 2 Element  2 = 70 Array 3 Element  2 = 1260
        Array 1 Element  3 = 16 Array 2 Element  3 = 63 Array 3 Element  3 = 1008
        Array 1 Element  4 = 14 Array 2 Element  4 = 56 Array 3 Element  4 =  784
        Array 1 Element  5 = 12 Array 2 Element  5 = 49 Array 3 Element  5 =  588
        Array 1 Element  6 = 10 Array 2 Element  6 = 42 Array 3 Element  6 =  420
        Array 1 Element  7 =  8 Array 2 Element  7 = 35 Array 3 Element  7 =  280
        Array 1 Element  8 =  6 Array 2 Element  8 = 28 Array 3 Element  8 =  168
        Array 1 Element  9 =  4 Array 2 Element  9 = 21 Array 3 Element  9 =   84
        Array 1 Element 10 =  2 Array 2 Element 10 = 14 Array 3 Element 10 =   28
        Array 1 Element 11 =  0 Array 2 Element 11 =  7 Array 3 Element 11 =    0
```

Sample Honors Output

```
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab6$ compile -DHONORS lab6.c
jimr@JimRXPS13:~/CS1050/FS2021/labs/lab6$ ./a.out
First Arrays:
        Array 1 Element  0 =  1 Array 2 Element  0 =  2 Array 3 Element  0 =    2
        Array 1 Element  1 =  4 Array 2 Element  1 =  7 Array 3 Element  1 =   28
        Array 1 Element  2 =  7 Array 2 Element  2 = 12 Array 3 Element  2 =   84
        Array 1 Element  3 = 10 Array 2 Element  3 = 17 Array 3 Element  3 =  170
        Array 1 Element  4 = 13 Array 2 Element  4 = 22 Array 3 Element  4 =  286
        Array 1 Element  5 = 16 Array 2 Element  5 = 27 Array 3 Element  5 =  432
        Array 1 Element  6 = 19 Array 2 Element  6 = 32 Array 3 Element  6 =  608
        Array 1 Element  7 = 22 Array 2 Element  7 = 37 Array 3 Element  7 =  814
        Array 1 Element  8 = 25 Array 2 Element  8 = 42 Array 3 Element  8 = 1050
        Array 1 Element  9 = 28 Array 2 Element  9 = 47 Array 3 Element  9 = 1316
        Array 1 Element 10 = 31 Array 2 Element 10 = 52 Array 3 Element 10 = 1612

First Arrays Reverse Multiplied:
        Array 1 Element  0 =  1 Array 2 Element  0 =  2 Array 3 Element  0 =   52
        Array 1 Element  1 =  4 Array 2 Element  1 =  7 Array 3 Element  1 =  188
        Array 1 Element  2 =  7 Array 2 Element  2 = 12 Array 3 Element  2 =  294
        Array 1 Element  3 = 10 Array 2 Element  3 = 17 Array 3 Element  3 =  370
        Array 1 Element  4 = 13 Array 2 Element  4 = 22 Array 3 Element  4 =  416
        Array 1 Element  5 = 16 Array 2 Element  5 = 27 Array 3 Element  5 =  432
        Array 1 Element  6 = 19 Array 2 Element  6 = 32 Array 3 Element  6 =  418
        Array 1 Element  7 = 22 Array 2 Element  7 = 37 Array 3 Element  7 =  374
        Array 1 Element  8 = 25 Array 2 Element  8 = 42 Array 3 Element  8 =  300
        Array 1 Element  9 = 28 Array 2 Element  9 = 47 Array 3 Element  9 =  196
        Array 1 Element 10 = 31 Array 2 Element 10 = 52 Array 3 Element 10 =   62

Second Arrays:
        Array 1 Element  0 = 22 Array 2 Element  0 = 84 Array 3 Element  0 = 1848
        Array 1 Element  1 = 20 Array 2 Element  1 = 77 Array 3 Element  1 = 1540
        Array 1 Element  2 = 18 Array 2 Element  2 = 70 Array 3 Element  2 = 1260
        Array 1 Element  3 = 16 Array 2 Element  3 = 63 Array 3 Element  3 = 1008
        Array 1 Element  4 = 14 Array 2 Element  4 = 56 Array 3 Element  4 =  784
        Array 1 Element  5 = 12 Array 2 Element  5 = 49 Array 3 Element  5 =  588
        Array 1 Element  6 = 10 Array 2 Element  6 = 42 Array 3 Element  6 =  420
        Array 1 Element  7 =  8 Array 2 Element  7 = 35 Array 3 Element  7 =  280
        Array 1 Element  8 =  6 Array 2 Element  8 = 28 Array 3 Element  8 =  168
        Array 1 Element  9 =  4 Array 2 Element  9 = 21 Array 3 Element  9 =   84
        Array 1 Element 10 =  2 Array 2 Element 10 = 14 Array 3 Element 10 =   28
        Array 1 Element 11 =  0 Array 2 Element 11 =  7 Array 3 Element 11 =    0

Second Arrays Reverse Multiplied:
        Array 1 Element  0 = 22 Array 2 Element  0 = 84 Array 3 Element  0 =  154
        Array 1 Element  1 = 20 Array 2 Element  1 = 77 Array 3 Element  1 =  280
        Array 1 Element  2 = 18 Array 2 Element  2 = 70 Array 3 Element  2 =  378
        Array 1 Element  3 = 16 Array 2 Element  3 = 63 Array 3 Element  3 =  448
        Array 1 Element  4 = 14 Array 2 Element  4 = 56 Array 3 Element  4 =  490
        Array 1 Element  5 = 12 Array 2 Element  5 = 49 Array 3 Element  5 =  504
        Array 1 Element  6 = 10 Array 2 Element  6 = 42 Array 3 Element  6 =  490
        Array 1 Element  7 =  8 Array 2 Element  7 = 35 Array 3 Element  7 =  448
        Array 1 Element  8 =  6 Array 2 Element  8 = 28 Array 3 Element  8 =  378
        Array 1 Element  9 =  4 Array 2 Element  9 = 21 Array 3 Element  9 =  280
        Array 1 Element 10 =  2 Array 2 Element 10 = 14 Array 3 Element 10 =  154
        Array 1 Element 11 =  0 Array 2 Element 11 =  7 Array 3 Element 11 =    0
```

Guidelines for Grading Lab 6
40 Points Possible

General

If your program does not compile or produce any input/output (I/O) because most of the source code is commented out then your lab will receive a grade of ZERO POINTS. Further, if your program does not actually follow the specifications, but merely prints out lines that make it appear to follow the specifications, you will receive a grade of ZERO POINTS. For partial credit your C program must not only compile but also produce some valid I/O that meets the lab specifications.

You program is expected to have a comment header at the top that includes your name, pawprint, the course you are taking, and the lab that you are solved (e.g., "Lab 6"). Your code should be nicely indented. You should include comments throughout your program that explain what you are intending to do. **You will lose up to 10 points if you do not meet these basic requirements**.

If you do **ANYTHING** in your main() other than declaring variables, calling printf(), or calling a function that you write (you may assign a variable to the return value of your function), you will lose 50% of the points you would have otherwise received. If you have any global variables, you will get zero points.

Non-Honors Rubric

5 points: Array 1 and Array 2 are initialized correctly and printed (even if not on the same line) for the "First Arrays".
5 points: Array 1 and Array 2 are initialized correctly and printed (even if not on the same line) for the "Second Arrays".
10 points: Array 1 and Array 2 are correctly multiplied to form Array 3 and Array 3 is printed (even if not on the same line) for the "First Arrays".
10 points: Array 1 and Array 2 are correctly multiplied to form Array 3 and Array 3 is printed (even if not on the same line) for the "Second Arrays".
10 points: Array 1, Array 2, and Array 3 are printed with corresponding elements on the same line.

Honors Rubric

5 points: Array 1 and Array 2 are initialized correctly and printed (even if not on the same line) for the "First Arrays".
5 points: Array 1 and Array 2 are initialized correctly and printed (even if not on the same line) for the "Second Arrays".
5 points: Array 1 and Array 2 are correctly multiplied to form Array 3 and Array 3 is printed (even if not on the same line) for the "First Arrays".
5 points: Array 1 and Array 2 are correctly multiplied using the "reverse multiply" algorithm to form Array 3 and Array 3 is printed (even if not on the same line) for the "First Arrays Reverse Multiplied".
5 points: Array 1 and Array 2 are correctly multiplied to form Array 3 and Array 3 is printed (even if not on the same line) for the "Second Arrays".
5 points: Array 1 and Array 2 are correctly multiplied using the "reverse multiply" algorithm to form Array 3 and Array 3 is printed (even if not on the same line) for the "Second Arrays Reverse Multiplied".

10 points: Array 1, Array 2, and Array 3 are printed with corresponding elements on the same line.