1. **(15 points)**Write a function to build and return the prime factorization of an unsigned integer, as a pointer to a dynamically allocated (from the heap) array of integers. The array should have value 0 as its last element. The prototype should be:

unsigned\* build\_factors(unsigned N);

For example, if N=600, the function should return a newly allocated array of size 7 with elements

{2, 2, 2, 3, 5, 5, 0}.

Note that the elements should be in order, except for the final 0. Since the size of the array will not be known until you determine the factors, you may use a large (say, 256 element) array to build the factors, and then either resize the array (if it was allocated from the heap) or copy the array. For the edge cases N=0 and N=1, return a 0 pointer.

1. **(10 points)**Write a function to display (to standard output) the elements of a 0-terminated array, not including the terminating zero. Elements should be separated by commas. The prototype should be:

void show\_array(unsigned\* A);

For example, if

A={2, 2, 2, 3, 5, 5, 0}

then the characters

2, 2, 2, 3, 5, 5

should be written. If A is a 0 pointer, then write nothing.

1. **(20 points)**Write a function that takes two 0-terminated arrays of factors, and builds and returns a new 0-terminated array of factors, that contains the common elements of the input arrays. More formally, if value i appears a times in the first array, and appears b times in the second array, then i should appear in the output array c times, where c is the minimum of a and b. The prototype should be:

unsigned\* build\_common(unsigned\* A, unsigned\* B);

For example, if

A={2, 2, 2, 3, 5, 5, 0};

B={2, 2, 5, 7, 0};

then the function should return a newly allocated array of size 4 with elements

{2, 2, 5, 0}.

Again, you may use a temporary array to build the result, and then copy or resize. If either input is a 0 pointer, return a 0 pointer.

1. **(5 points)**Write a main program that prompts the user for two integers M and N, and displays (to standard output) their individual factors and their common factors. For example:

Enter two integers M, N:

600, 140

600 has factors: 2, 2, 2, 3, 5, 5

140 has factors: 2, 2, 5, 7

600 and 140 have common factors: 2, 2, 5

Collect all this together into a single file, factors.c, complete with documentation including your name, and upload it to Canvas. Your code will be compiled and tested on pyrite.