CS3024 Home Assignment 1

(20 points)

Write a C++ subroutine that given two arrays of integers, merges them into a new array. You'll need also to design a subroutine to print an array to show the inputs and the output:

```
void show(int a[], int len);
```

Your solution should be a subroutine with the following prototype:

```
int * append(int a[], int length1, int b[], int length2);
```

It receives two arrays as input (the pointer to the first array element and the length of array), dynamically allocates memory for the resulting array, and returns a pointer to the first element of resulting array. Notice that the **length1** and **length2** may be smaller than actual size of array **a** or **b**, correspondingly. That means that append operation may be performed only on the first elements of an array. For example, the length may be 0.

You may write the following code to demonstrate your solution.

```
const int len1 = 5;   // 5 is just an example
int a1[len1] ;

const int len2 = 3;  // 3 is just an example
int a2[len2] ;
```

Initialize the arrays using random numbers or array initialization list.

```
// print the contents of a1 and a2 before the calls to append()
show(a1, len1);
show(a2, len2);

int * result;

// first test case
result = append(a1, len1, a2, len2);
show(result, len1 + len2); // print the contents of result

 // more test cases
```

In order to test your solution, call **append()** with different array lengths, one for each first and second argument:

- array of length 1,
- array of length 0 (in other words, make sure that your program does not crush when called with the array of length 0 input, and does not alter array's contents),
- array of a small length larger than 1.

Please notice that you don't need to declare separate arrays for testing, you can use the same arrays **a1** and **a2**, just using different lengths when calling **append().**

Missing a required test case will reduce your grade (at least 9 test cases are needed to cover all possible combinations of cases described above).

Your C++ code should have a header comment including your name, date, short description of the task, and file name.

Use indentation and blank lines to make your code more readable. Add comments to explain the purpose of major code segments. Read the article *p57-green-Coding-Standards.pdf* posted on Sakai for details on making the source code more readable.

Submit electronically (to maugusto@nps.edu) the source code file with the set of test cases you have used to test your program written in the body of **main()** function. Attach the .cpp file to your email. **Include your last name** into the source file name.

If you don't receive a confirmation from me by the next day, please submit again to avoid problems of the type "I've submitted it, but…" I don't require the hard copy of the source code this time.

The deadline is **Monday, October 30, 2017, midnight.**