

**Simple Array Manipulation in C****Circular Left Shift****Assignment**

For this assignment, you will complete the implementation of a C function that takes an array of integer values and performs a 1-cell circular left shift on the contents of the array:

Original array contents:	18	31	8	76	42	17
Modified array contents:	31	8	76	42	17	18

So, the original first element moves to the end of the array (hence, circular) and every other array element moves one spot to the left.

You will complete the implementation of the following C function:

```

/*  shiftArray()
 *
 *  Performs a left circular shift of 1 position on array.
 *  Pre:
 *      Size > 0
 *      List[0:Size-1] have been initialized
 *  Post:
 *      List[k] == original List[(k+1) % Size] for k = 0:Size-1
 */
void shiftArray(int List[], int Size) {
    . . .
}

```

Your implementation may, and should, assume that the stated preconditions are met. There is no way for you to validate them in C. Assuming the preconditions are satisfied, your implementation should guarantee that the stated postcondition will be satisfied.

More precisely, you will complete the implementation of the short C program that is supplied later in this specification (and on the course website). The supplied program provides code to read test data from a text file, and to log results to a second text file. Sample test data files are provided on the course website.

The supplied program will compile and run as given, but it will not modify the given array. You must modify the program, as indicated, so that the function modifies the array as described above. There are two restrictions: your function implementation may not write any output, and it may not declare any local arrays except for the parameter `List[]`.

You should test your solution thoroughly; there is no guarantee that the supplied test data covers all cases.

You will submit your completed `.c` file, making changes only where indicated in the supplied code.

Your submission will be compiled, tested and graded according to how many cases your solution handles correctly.

You will be allowed up to ten submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every test case you can think of. If you do not get a perfect score, analyze the problem carefully and test your fix with the test data shown in the Curator grade report, before submitting again. The highest score you achieve will be counted.

The *Student Guide* and other pertinent information, such as the link to the proper submit page, can be found at:

<http://www.cs.vt.edu/curator/>

**Supplied code:**

```

/* CS 2505 Summer I 2010
 * Supplied framework for circular left array shift project.
 *
 * Student:  <ENTER YOUR NAME HERE>
 * PID:      <ENTER YOUR EMAIL PID HERE>
 */
#include <stdio.h>
#include <stdlib.h>

void shiftArray(int List[], int Size);
int  readData(int **A);
void printList(int List[], int Size, FILE *Out);

int main(int argc, char** argv) {

    int *List;           // pointer for list
    int Size = readData(&List); // create array from input file

    FILE *Log = fopen("Results.txt", "w"); // try to open log file
    if ( Log == NULL ) { // make sure it was created
        printf("Error:  could not open results file.\n");
        return 1;
    }

    fprintf(Log, "Original list:\n"); // log original list
    printList(List, Size, Log);

    shiftArray(List, Size); // perform circular left shift

    fprintf(Log, "Modified list:\n"); // log modified list
    printList(List, Size, Log);

    fclose(Log); // close log file
    return 0;
}

/* shiftArray()
 *
 * Performs a left circular shift of 1 position on array.
 * Pre:
 *     Size > 0
 *     List[0:Size-1] have been initialized
 * Post:
 *     List[k] == original List[(k+1) % Size] for k = 0:Size-1
 */
void shiftArray(int List[], int Size) {

    // DO NOT MODIFY ANY CODE ABOVE THIS LINE!!

    // Implement function according to the given pre- and post-conditions here.

    // DO NOT MODIFY ANY CODE BELOW THIS LINE!!
}

```

```
/* printList()
 * Pre:
 *     Size > 0
 *     List[0:SZ-1] have been initialized
 *     Out is opened on a text file
 * Post:
 *     List[0:SZ-1] have been printed to file on a single line
 */
void printList(int List[], int Size, FILE *Out) {

    for (int pos = 0; pos < Size; pos++) {
        fprintf(Out, "%5d", List[pos]);
    }
    fprintf(Out, "\n");
}

/*
 * readData()
 * Pre:
 *     Data.txt exists and holds at least Size integer values, one per line
 * Post:
 *     An array List[] has been allocated and the target of the parameter A
 *     set to point to that array.
 *     List[0:SZ-1] have been initialized to the first Size values in the file
 * Returns:
 *     Number of values read from the file.
 */
int readData(int **A) {

    FILE *In = fopen("Data.txt", "r");        // read number of elements
    int Size;
    fscanf(In, "%d\n", &Size);

    *A = malloc( Size * sizeof(int) );        // allocate array
    int *p = *A;                             // get temp pointer into array

    // read and count array elements from the file;
    // stop if array is filled or the file contains fewer elems than expected;
    // scanf() returns the number of values that were read on each call
    int numRead = 0;
    while ( numRead < Size && fscanf(In, "%d\n", p) == 1 ) {

        ++numRead;
        ++p;
    }

    fclose(In);
    return numRead;
}
```

**Pledge:**

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the submitted file:

```
// On my honor:  
//  
// - I have not discussed the C language code in my program with  
// anyone other than my instructor or the teaching assistants  
// assigned to this course.  
//  
// - I have not used C language code obtained from another student,  
// or any other unauthorized source, either modified or unmodified.  
//  
// - If any C language code or documentation used in my program  
// was obtained from another source, such as a text book or course  
// notes, that has been clearly noted with a proper citation in  
// the comments of my program.  
//  
// - I have not designed this program in such a way as to defeat or  
// interfere with the normal operation of the Curator System.  
//  
// <Student Name>
```

**Failure to include this pledge in a submission is a violation of the Honor Code.**