



GNG1106 - Lab 7

Objectives

Practice the use of:

1. Accessing variables in an array by indexing
2. Functions with pointers in their parameters ("pass-by-reference" functions)

Introduction

Circular Shift Left (CSL) is an operation in combinational mathematics that rearranges a tuple of entries by moving the first element to the last position, while shifting all other elements to the neighbouring vacant position. For example, if we repeatedly apply CSL 4 times to the 4-tuple (1, 2, 3, 4) it will successively give:

1. (2, 3, 4, 1)
2. (3, 4, 1, 2)
3. (4, 1, 2, 3)
4. (1, 2, 3, 4)

Consult this link for more details on circular shift: https://en.wikipedia.org/wiki/Circular_shift

Instructions

Pre-Lab Submission (20%): Ensure you have submitted your pre-lab before attending the lab session.

Deliverable 1 (20%): Write a C program to perform a CSL operation over an array of integers of length N. N should be defined as a symbolic constant equal to 100, and the array entries should be initialized from 1 to 100 ($\text{array}[i]=i+1$). Your program should also output on the screen the shifted array, if successful, the output should look similar to this:

```
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85
86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 1
```

Be careful with indexing and avoid any illegal memory access!

Deliverable 2 (20%): Upgrade the basic version CSL in deliverable 1 to allow s CSL operations over the array entries. Your program should ask the user to enter the number s of CSL operations to be performed, where s is a positive integer.

For example, if the user enters 5, your program should display:

```
6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88
89 90 91 92 93 94 95 96 97 98 99 100 1 2 3 4 5
```

Deliverable 3 (40%): Upgrade deliverable 2 code to have 3 functions with the following prototypes:

```
1. void get_s(int *s); //get user's input for the number of CSL
   operations
2. void CSL(int *a, int *s); // perform s number of CSL operations to
   the array addressed by a.
3. void prtArray(int *a); // print the content of the array addressed
   by a.
```

The above functions should be called by the main function in this order.

Check out and Submission

You must check out with your TA before submitting the deliverables. During the check out, your TA may inspect your work and to-be-submitted deliverables, and ask you questions to further check your understanding. At the end of the check out, your TA will give you an initial mark for the in-lab component of this lab and let you know. You must then submit the deliverables before the due time of this lab. While this initial mark is likely to be the final, your TA reserves the right to reduce this initial mark after checking more carefully the deliverables you submit

Grading Criterion

- **Correctness (60%):** Correct syntax, logic and execution.
- **Style (40%):** Descriptive variables names and appropriate indentation (especially those in the loops), function calls, prototyping, parameters passing, and array indexing.