

Lab 7 (19th Jan 2022)

Problem 1

- Implement both approaches to Hackathon 1's Problem 3 from the solution document.
 - Use call by reference to call your function on the input array.
 - Make your program indicate a mismatch location if any.

Problem 2*

- Input: $n \in \mathbb{N}$
- Goal: Print all $n!$ permutations of the numbers $1, 2, \dots, n$.
- Example:
 - Input: 3
 - Output:
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1
- *Hint: Understand how you write down the permutations on paper. Then write the algorithm, followed by translating it to C code.*

Problem 3

- Input: $n \in \mathbb{N}$, followed by a sequence $a_1, a_2, \dots, a_n \in \mathbb{Z}$.
- Goal: Store the input sequence in an array. Then reverse the sequence within the same array.
 - Use dynamic memory allocation to create an array on the heap.
 - Call a function that reverses the array.
 - Print the array from beginning to end.

Problem 4

Create two `int` variables, say `var1` and `var2`, and create pointers to these variables. Let `ptr1` be a pointer to `var1`, and `ptr2` be a pointer to `var2`. Write a function called "foo" such that after foo is called, `ptr1` will point to `var2` and `ptr2` will point to `var1`.

(Hint: A pointer that points to a pointer to int has type `int **`)