# SCC.111: ARRAY "FUN"

## THE GOAL

This week's 111 lab aims to:

- [ ] Practice working with arrays in C
- [ ] You'll also get to practice your 'computational problem solving', working from a high level problem specification to a C solution
- [ ] There is a main problem with a few sub-tasks of increasing difficulty. There is also a more advanced 'hacker edition', of course ;)

## PROBLEM STATEMENT

The 'high score table' is the absolute classic addition to video games. First seen in Asteroids in 1979(!), the idea is simple. Depending on your level of achievement your score is presented in a 'ranked list', with the highest score - not surprisingly - at the top!


Figure 1:

You are tasked with creating a high score table to hold the top 10 scores of some ficticious game. Ideally, your solution should add a score to the table, maintaining the ordering property that the highest score is at the top, in descending order to the lowest of the 10 highest scores last.

The output from my program looked like this by the end of the task:

```
Enter high score (0 to exit):400
*** HIGH SCORE TABLE ***
 1 - 500
 2 - 400
 3 - 350
 4 - 250
 5 - 200
 6 - 150
 7 - 100
 8 - 90
 9 - 85
10 - 80
Enter high score (0 to exit):0
Highest score = 500
Lowest score = 80
```

## PROBLEM 1: BASIC GROUNDWORK

Step 1, as ever, is to think carefully about how your algorithm will work. **Design on paper highly recommended!**

Incremental steps would be:

1. start a new program e.g. ‘ `highscore.c` ’.
2. declare a suitable array for your high score table.
3. Ensure that you can enter, store data in, and print out your high score table. You can read a number as input using `scanf` :

```
scanf("%d", &score);  // score is an int, %d means read an 'int / decimal'
```

## PROBLEM 2: ADDING SCORES

Note that by convention high score tables are initially all set to zero. As players finish their games, the score gets added in the appropriate position in the table (or not at all if they're not in the top 10!).

1. Create a basic user interface, where the user enters their score.
2. The score should get added to the 'next free slot' in the high score table, when you have no more room (all 10 slots full an error message should be displayed instead)
3. Find the largest and smallest scores in the list and print these out at the end
4. Add the logic so if the player's score is lower than the lowest score in the table, and there's no more room, the score is not added to the table

*Important note: the pseudocode assumes the array is indexed from 1, not zero biased (from 0) as in C arrays!*

***Don't forget to show us your solutions!***

# SCC121: FUNCTIONS

This week's lab activities will cover the topic of *Function*. A function, at its core, is a relationship that maps each input to a unique output, akin to how algorithms process information. This seemingly simple idea forms the bedrock of complex algorithms, data processing, and even the design of new programming languages. As you embark on this exercise sheet, you'll delve deep into the algebraic concept of functions, exploring their intricacies and uncovering their profound significance in computational logic and programming. Mastering these foundational concepts will not only bolster your mathematical prowess but also elevate your computational thinking, preparing you for advanced topics in Computer Science.

## THE GOAL

Using the knowledge of the lecture material from Week 3, you are expected to work through the following questions. These problems will equip you with the essential knowledge and skills to analyze functions, make connections, and solve complex problems. You should work out the answers *with a pen and paper*!

1. Let A = {a, b, c, d, e} and B = {1, 2, 3, 4} with f(a) = 2, f(b) = 1, f(c) = 4, f(d) = 1 and f(e)= 1.

   - Is f a function? why?
   - What is the domain of this function?
   - What is the co-domain?
   - What is the range of this function?

2. Let the function f(x) = 3x - 3 on the set of real numbers. Find its inverse.

3. Let A = {1, 2, 3}, and B = {1, 2, 3, 4}, and $f : A \to B$, $g : A \to B$ by:

   - f(1) = 4, f(2) = 2, f(3) = 1.
   - g(1) = 4, g(2) = 4, g(3) = 1.

Which of these functions is injective?

4. Let A = {1, 2, 3, 4}, and B = {1, 2, 3, 4} and $f : A \to B$, $g : A \to B$ by:

- f(1) = 4, f(2) = 2, f(3) = 1, f(4) = 3.
- g(1) = 4, g(2) = 3, g(3) = 1, g(4) = 1.

Which of these functions is surjective?

5. Let f(x) = 3x + 6 on the set of real numbers.

   - Is f a bijective function?
   - If so, what is its inverse?

6. Let M and N be relations from A to B, with A = {a, b, c, d}, and B = {s, t, u}, defined as follows:

   - M = {<a, t>, <b, s>, <c, s>, <d, u>}
   - N = {<a, s>, <b, t>, <c, s >, <a, u>, <d, u>}

   Which one of the following statements is true?

   - M and N are functions
   - M is a function and N is not a function
   - M and N are not functions
   - none of these

7. Let f be a function on the set N of natural numbers, f: N → N, defined by f(x) = 2x + 3. Is function f:

   - injective
   - surjective
   - bijective
   - none of the above

8. Let f: Z→ Z be defined by f(x) = 2x + 1

   - what is the domain, codomain, range of f?
   - is f one-to-one (injective)?
   - is f onto (surjective)?
   - if f is bijective, what is its inverse?

9. Let f_1 and f_2 be two functions from A to B such that

   f_1(x) = x_2 and f_2(x) = $x - x^2$.

   - What is the sum function f1+ f2?
   - What is the product function f1x f2?

10. Let f and g be two functions from the set of integers to the set of integers, defined by $f(x) = 2x + 3$ and $g(x) = 3x + 2$

    - What is the composition of f and g?
    - What is the composition of g and f?

11. Let $f(x) = x^2 + 1$ and $g(x) = x + 2$ be two functions from the set of integers to the set of integers. Find the following functions:

- $f + g$
- $(f + g)\ (1)$
- $f \times g$
- $(f \times g)\ (0)$
- $f \circ g$
- $(f \circ g)\ (1)$
- $g \circ f$
- $(g \circ f)\ (2)$

School of Computing & Communications | Lancaster University

# SCC.131: BOOLEAN LOGIC

In this week's lab activities, we will practice Boolean Logic. In particular, we will practice minimizing Boolean functions using Boolean algebra and Karnaugh maps.

**The Goal**

By applying the knowledge of the lecture materials from Week 3, you are expected to work through the following questions. You should work out the answers *with a pen and paper* - **no calculators nor computers**!

By the end of this task you should be able to:

- [ ] Be familiar with Boolean algebra and Karnaugh maps.
- [ ] Simplify Boolean functions using Boolean algebra.
- [ ] Simplify Boolean functions using Karnaugh maps.

**Today's Tasks**

This is a worksheet for the SCC.131 week 4 lab practical. Work through the following questions and ask us if you need help.

1. Express the following statements in Boolean algebra:
   - Either A is true or B is true.
   - A is not true.
   - Neither A nor B is true.
   - B is true, but not A.
2. Simplify the following Boolean expressions:
   - A+A'
   - A+A
   - A' + A'A
   - A' + AB
3. Use a truth table to evaluate the following expressions:

- A + B
- A' + B'
- A + A'
- A'B'

4. Minimise the following Boolean functions using Boolean algebra:

- F = AB + A'B + AB'
- F = ABC + AB'C + ABC' + AB'C' + A'BC + A'B'C

5. Minimise the following Boolean functions using Karnaugh maps:

- F = AB + A'B + AB'
- F = ABC + AB'C + ABC' + AB'C' + A'BC + A'B'C
- F = A'BCD + A'B'CD + A'BC'D + A'B'C'D + ABC'D' + AB'C'D' + ABCD' + AB'CD'

# HACKER EDITION

## SCC.111: ARRAY "FUN"

Add the ability to add the score in the correct location in the high score table, so the table is always maintained in descending sorted order from highest to lowest score.

If you'd like, create a version which implements a sort algorithm instead. For example, using 'bubble sort', as outlined below:

'Bubble sort' is a well-known sort algorithm that puts a list of items into ascending order by 'bubbling' the smallest item to the top (first array position), then the next smallest to the next slot (second position) etc. It stops *when no swaps are made* (all items sorted), this is an important optimisation particularly for pre-sorted data.

Pseudocode is as follows (see animation):

```
for i = 1:n,
    swapped = false
    for j = n:i+1,
        if a[j] < a[j-1],
            swap a[j,j-1]
            swapped = true
    → invariant: a[1..i] in final position
    exit if not swapped
end
```

## SCC.121: FUNCTIONS

1. Let A = {1, 2, 3, 4, 5} and B = {a, b, c, d, e} and the function shown in the arrow diagram:

- define the relation from A to B as reflected in the diagraph
- is this a function? why?
- what is image of 3?

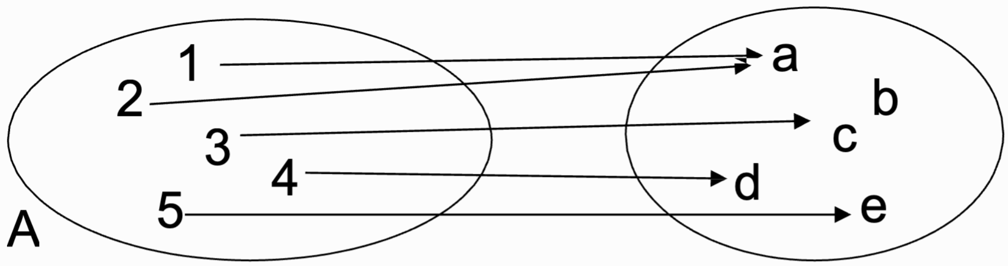


Figure 2:

- what is the preimage of a?

1. Let A = {1, 2, 3, 4, 5} and B = {a, b, c, d, e} and the function shown in the arrow diagram shown in Exercise 12.

- is this function surjective? why?
- is it injective? why?

2. Let A = {1, 2, 3, 4, 5} and B = {a, b, c, d, e} and the function shown in the arrow diagram shown in Exercise 12.

- how would you redefine the codomain so that the function becomes surjective?
- how would you redefine the set of ordered pairs so that the function becomes injective?
- define the bijective new function that you constructed
- find its inverse.

# SCC.131: BOOLEAN LOGIC

1. Prove the following identities using Boolean Logic algebraic manipulations:

- A + A’B = A + B
- A + AB = A

2. The Consensus Theorem is a fundamental identity in Boolean algebra. It provides a way to eliminate a redundant term in specific Boolean expressions. The theorem states that:

$$AC + A'B + BC = AC + A'B$$

- Prove the theorem using boolean algebra.
- Prove the theorem using a Karnaugh Map.

3. Examine the provided Karnaugh Map (K-Map), identify the mistake, and correct it to achieve the simplified Boolean expression. Consider a 3-variable K-Map for the function F(A,B,C).

F(A,B,C)= A'B'C' + AB'C + A'BC + A'BC' + ABC'

| **A BC** | **00** | **01** | **10** | **11** |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 |

The correct boolean expression is F(A,B,C) = A'B + BC' + A'C'. Analyze the provided Karnaugh map. Is the simplified expression correct? Can you find the mistake?