Pseudocode – Representing algorithms

- Is it a programming language?
  - No, but it is something close.
  - It is not close enough to any particular language to give anyone an unfair advantage.
  - Provides a language independent way to describe an algorithm.
  - Formal enough to convert into any programming language.
  - Let's see some pseudocode writing rules!
  - Very important - be consistent!

Let's take an example – Insertion sort

```
INSERTION-SORT(A)
    for j <- 2 to length[A]
        key <- A[j]
        i <- j - 1
        while i > 0 and A[i] > key
            A[i+1] <- A[i]
            i <- i - 1
        A[i+1] <- key
```

```
INSERTION-SORT(A)
    for i <- 1 to length[A]
        key <- A[i]
        j <- i
        while j > 0 and A[j-1] > key
            A[j] <- A[j-1]
            j <- j - 1
        A[j] <- key
```

How does insertion sort work?

Is there any difference between the versions on the previous slide?

How does this “code” differ from Python or C++?

Pseudocode conventions - Assignment

- Pseudo-code uses ← for assignment instead of '=' used by most programming languages.
- '=' is used for comparison instead of '=='
- Multiple assignment is achieved with: a←b←0 (a and b are variables).
- Variables are considered local; for global we should explicitly state so.
- Arrays in pseudocode sometimes start at 1, unlike most programming languages which start at 0. Accessing array elements - standard way A[i], where i is the potion of the element in the array.
- Range of an array: A[1..j] - j elements.
- Depends on the author -> Consistency is key!

Pseudocode conventions - Indentation

- Code blocks are defined with indentation, this will be familiar to Python programmers
- Example:

```
if (proposition p is true)
    this gets done if p is true
    this also gets done if p is true
this gets done regardless of whether or not p is true
```

- This also applies to loops and functions.

Pseudocode conventions – Loops and conditionals

- Loops, such as while and for retain their commonly understood meanings, as do if and else statements.
- Loop counters retain their value after the loop has finished -> the value it will have after exiting a for loop is the first value that exceeded the loop bound.
- To and Downto keywords.
- Example: for i <- 1 to length[L] the value of i will be length[L] + 1 (or n + 1)

Pseudocode conventions - Functions

- Capitalise function name, include parameters in brackets.
- Example: DO-STUFF(THING1, DATE)

Pseudocode conventions - Boolean operators

- And, Or logical operations.
- Short circuiting - do you remember from last year?
- x and y - we first evaluate x
- What happens if x is False?
- What happens if x is True?
- Allows us to work with things like (x ≠ NIL and x.f = y).

Pseudocode conventions - Miscellaneous

- If you want to add comments to the pseudocode, use the // symbol.
- We use return statements - the difference compared to traditional programming languages is that we can return multiple values in a single return statement.
- Write only one statement per line.
- It has to be language independent - common mistake is writing Python code as due to its high level style it can be impulse.

Pseudocode examples – GCD and SMC

- Write the pseudocode for an algorithm that computes the greatest common divisor (gcd) and the smallest common multiple (smc) of two natural numbers, a and b.
- What is a GCD? The gcd of two or more integers, when at least one of them is not zero, is the largest positive integer that divides the numbers without a remainder. For example, the gcd of 8 and 12 is 4.
- What is a LMC? The smc of two integers a and b, is the smallest positive integer that is divisible by both a and b. For example, the lmc of 4 and 6 is 12.

Pseudocode examples – GCD and SMC

- To compute gcd(48,18), divide 48 by 18 to get a quotient of 2 and a remainder of 12. Then divide 18 by 12 to get a quotient of 1 and a remainder of 6. Then divide 12 by 6 to get a remainder of 0, which means that 6 is the gcd.
- gcd(a,0) = a
- gcd(a,b) = gcd(b,a mod b)
- Euclid's algorithm

Pseudocode examples – GCD and SMC

```
COMPUTE_GCD_LMC(a,b)
    a1 <- a
    b1 <- b
    rem <- a1 mod b1
    while (rem != 0)
        a1 <- b1
        b1 <- rem
        rem <- a1 mod b1
    gcd <- a1
    lmc <- a mult b div gcd
```

Pseudocode examples – Palindrome

- Read from the keyboard a natural number n. Check if this number is palindrome.
- What is a palindromic number?
- A palindromic number is a number that remains the same when its digits are reversed. For example, 27472.

Pseudocode examples – Palindrome

```
CHECK_PALIDROME(n)
    r <- reversed n
    for i <- 0 to (n.length + 1) div 2
        if r[i] ≠ n[i]
            return FALSE
    return TRUE
```