Intro to pseudocode!

Conditional Statements – IF, THEN, ELSE

Runs one chunk of code depending on a condition, then skips the other chunks of code.

```
input  a, b
if a ≥ b then
    print a
else
    print b
end if
```

If a and b were inputted as 5 and 12 respectively, what's the output?

```
input  mark
if mark ≥ 90 then
    print 'A'
else if mark ≥ 75 then
    print 'B'
else if mark ≥ 60 then
    print 'C'
else if mark ≥ 50 then
    print 'D'
else
    print 'E'
end if
```

If $mark$ was inputted as 68, what's the output?

Loops – FOR

Repeats the code inside the loop a certain number of times from a starting number to a finishing number.

Algorithm:

```
sum ← 0
for i from 1 to 4
    sum ← sum + i²
end for
print sum
```

Desk check:

Algorithm:

```
input n
sum ← 0
for i from 1 to n
    sum ← sum + 1/i³
end for
print sum
```

Desk check:
($n = 3$)

Loops – WHILE

Repeats the code inside the loop while a condition is met.
Once the condition is not met, the loop stops.

Algorithm:

```
count ← 0
remainder ← 72
while remainder ≥ 14
    count ← count + 1
    remainder ← remainder − 14
end while
print count, remainder
```

Desk check:

```
A ← 1.3
while A³ > 2.01 or A³ < 1.99
    A ← 0.5 × (A + 2/A²)
    print A, A³
end while
```

Desk check:

For each algorithm, have a go at desk checking it:

```
define factorial(n):
  product ← 1
  for i from 1 to n
      product ← product × i
  end for
  return product
```

```
sum ← 0
for i from 1 to 10
    sum ← sum + 1/factorial(i)
end for
print sum
```

Run for when a = 1 and b = 3

```
input a, b
while a + b < 20
    b ← b + 2a
    a ← a + 2
end while
print a, b
```

```
sum ← 0
for i from 1 to 5
    sum ← sum + i
end for
print sum
```

Run for when x = 3

```
total ← 0
for i from 1 to x
    total ← total + i
end for
print total
```

```
a ← 1
b ← 2
for n from 1 to 5
    c ← b − n × a
    b ← a
    a ← c
    print n, a, b, c
end for
```

```
x ← 1
while x² < 1000
   x ← x + 1
   print (x − 1)²
end while
```

```
for i from 1 to 5
   if i² < 10
      print i²
   else
      print i² + 2
   end if
end for
```

```
sum ← 0
for i from 1 to 1000
   outcome ← 0
   count ← 0
   while outcome ≠ 6
      outcome ← randominteger(1, 6)
      count ← count + 1
   end while
      sum ← sum + count
   end for
   print sum/1000
```

Nested Loops – Loops inside loops!

```
c ← 0
for a from 1 to 2
    for b from 1 to 3
        c ← c + 1
    end for
end for
```

```
c ← 0
for a from 1 to 3
    for b from 1 to 2
        c ← c + ab
    end for
end for
```