Pseudocode Snippets

- running total within a loop — adding up a list of values
- input validation — checking that input is acceptable
- traversing a 1D array — accessing each element of an array from first to last

Running total within a loop: example 1 (fixed loop)
This program is used to calculate the sum of a known number of values entered by the user one at a time.

```
1. SET total to 0

2. Fixed loop FROM 1 TO 10 DO

  3.  RECEIVE number FROM KEYBOARD

  4. SET total TO total + number

5.END Fixed loop
```

Running total within a loop: example 2 (conditional loop)
This program is used to calculate the sum of an unknown number of values entered by the user one at a time.

```
1.SET total to 0

2.SET choice to "yes"

3.Conditional Loop: while choice = "yes" THEN

   4.RECEIVE number FROM KEYBOARD

   5.SET total TO total + number

   6.SEND "Do you wish to enter another value - yes/no" TO DISPLAY

   7.RECEIVE choice FROM KEYBOARD

8.End Loop
```

Input validation: example 1 (while loop)
This program is used to obtain a value between 10 and 20 inclusive.

```
1. RECEIVE number FROM KEYBOARD
2. Conditional Loop: WHILE number < 10 OR number > 20 DO
    3. SEND "Error, please enter again" TO DISPLAY
    4. RECEIVE number FROM KEYBOARD
5. END WHILE loop
```

Traversing a 1D array: example 1 (fixed loop)
This program is using a loop to access each element of an array, for the purposes of processing the data in the array.

```
1.DECLARE allScores INITIALLY [ 12,34,23,54,32,67,26,23 ]
2.FOR counter FROM 0 TO 7 DO
  3.IF allScores[counter] >= 50 THEN
     4.SEND "Great Score" & allScores[counter] TO DISPLAY
5.END IF
6.END FOR
```

Traversing a 1D array: example 2 (fixed 'for each' loop with running total included) This program is using a loop to access each element of an array, for the purposes of processing the data in the array.

```
1.DECLARE allScores INITIALLY [ 12,34,23,54,32,67,26,23 ]
2.DECLARE total INITIALLY 0
3.DECLARE counter INITIALLY 0
4.FOR EACH FROM allScores DO
  5.SET total TO total + allScores[counter]
6.SET counter TO counter + 1
7.END FOR
```

More Pseudocode Examples

Input validation: example 1 (while loop)

This program is used to obtain a value between 10 and 20 inclusive.

```
1. Get number (integer) from user
2. Start Conditional Loop. WHILE number < 10 OR number > 20 DO
3.   SEND "Error, please enter again" TO DISPLAY
4.   Re-enter. Get number (integer) from user
5. END Loop
```

Input validation: example 2 (until loop)

This program is used to obtain a value between 10 and 20 inclusive.

```
1. Start Conditional Loop
2. Get number (integer) from user
3.   IF number < 10 OR number > 20 THEN
4.     SEND "Error, please enter again" TO DISPLAY
5.   END IF
6. LOOP UNTIL number >= 10 AND number <= 20
```

Running total within a loop: example 1 (fixed loop)

This program is used to calculate the sum of a known number of values entered by the user one at a time.

```
1. Set total = 0
2. Fixed Loop to repeat 10 times
3.   Get number (integer) from user
4.   SET total TO total + number
5. END FOR
```

Traversing a 1D array: example 1 (fixed loop)

This program is using a loop to access each element of an array, for the purposes of processing the data in the array.

```
1. DECLARE allScores INITIALLY [ 12,34,23,54,32,67,26,23 ]
2. Start Fixed Loop to repeat 8 times (counter)
3.  IF allScores[counter] >= 50 THEN
4.    Print "Great Score" & allScores[counter] TO DISPLAY
5.  END IF
6. END Loop
```

Example IF... ELSE statement

```
1. IF mark < 50 THEN
2.    SET grade TO "Fail"
3. ELSE
4.    SET grade TO "Pass"
5. END IF
```

Nested IF example

```
1.  IF mark>=70 THEN
2.    SET grade=A
3.  ELSE
4.    IF mark>=60 THEN
5.      SET grade=B
6.    ELSE
7.      IF mark>=50 THEN
8.        SET grade=C
9.      ELSE
10.       SET grade=D
11.     END IF
12.   END IF
13. END IF
```

Pseudocode: example 2

This algorithm adds up the length of the tracks on a CD.

Algorithm

```
1  Initialise total length
2  Get valid number of tracks from user
3  Start fixed loop for each track
4     Get title and track length from user
5     Add track length to total
6  End fixed loop
7  Display track titles and track lengths
8  Display total length
```

Refinement

```
2.1  Start conditional loop
2.2      Get number of tracks from user
2.3      If number of tracks is not valid display error message
2.4  Repeat until the number of tracks entered is between 1 and 20
     inclusive

4.1  Get track title and store in names array
4.2  Get track length and store in length array

5.1  Add track length to total length

7.1  Start fixed loop for length of names array
7.2      Display "The name of track", counter, "is", track name
7.3      Display "The length of track", counter, "is", track length
7.4  End fixed loop

8.1  Display "The total length of the tracks is", total length
```

Pseudocode: example 1
This algorithm calculates the volume of a swimming pool.

Algorithm

```
1    Ask user to enter dimensions of a swimming pool in metres
2    Calculate volume of pool
3    Display message stating the volume of the pool
```

Refinement

```
1.1  Ask user to enter length of pool
1.2  Ask user to enter width of pool
1.3  Ask user to enter depth of pool

2.1  Volume is calculated as length * width * depth

3.1  Display "The volume of the pool is", volume
```

- A program is to be developed to create usernames for a class of twenty pupils
- The program will ask a teacher to enter the first name, surname and age of each pupil.
- The age entered must be between five and eighteen.
- The program should output a list of usernames for the teacher.

Main steps (algorithm)

```
1 Initialise username
2 Start fixed loop for twenty pupils
3    Get first name and surname from user
4    Get valid age from user
5    Generate username
6 Display "Username", index, "is" username
7 End fixed loop
```

Refinement

```
3.1 Get first name and store in first name array
3.2 Get surname and store in surname array

4.1 Get age and store in age array
4.2 While age is less than 5 or higher than 18 start conditional loop
4.3   Display error message "Invalid age, enter between 5 and 18 inclusive"
4.4   Get age and store in age array
4.5 End conditional loop

5.1 Concatenate first name, surname and age and store in username array
5.2 Start fixed loop from 0 to index -1
5.3   If username equals stored username
5.4     Add 1 to age
5.5     Concatenate first name, surname and age and store in username array
5.6   End If
5.7 End fixed loop
```