Extension activity the CASE statement

Sometimes we have a situation where there are many outcomes to a decision.

Consider a student exam mark.
Less than 20 is a ‘F’. 20-29 is a ‘E’, 30-49 is a ‘D’, 50-64 is a ‘C’, 65-79 is a ‘B’, 80 and over is an ‘A’.

The teacher needs a program to read in the mark and output either ‘A’, ‘B’, ‘C’, ‘D’, ‘E’ or ‘F’.

A CASE statement is needed.

```
CASE mark OF
  0 to 19 WRITE ‘F’
  20 to 29 WRITE ‘E’
  30 to 49 WRITE ‘D’
  50 to 64 WRITE ‘C’
  65 to 79 WRITE ‘B’
  80 to 100 WRITE ‘A’
ELSE
  WRITE ‘Invalid mark’
END
```

The Input-Process-Output (IPO) chart is:

Input	Processing	Output
mark	In the case of mark being 0 to 19 WRITE ‘F’ 20 to 29 WRITE ‘E’ 30 to 49 WRITE ‘D’ 50 to 64 WRITE ‘C’ 65 to 79 WRITE ‘B’ 80 to 100 WRITE ‘A’ Else Write ‘Invalid mark’ End	grade

Caption: An IPO chart for the Grade algorithm.

The algorithm, Pseudocode, Pascal and Python programs.

Algorithm

```
Write ‘A simple program to grade a mark.';
Write ‘Please give the mark.’
Get the mark
In the case of mark being
  0 to 19 Write ‘F’
  20 to 29 Write ‘E’
  30 to 49 Write ‘D’
  50 to 64 Write ‘C’
  65 to 79 Write ‘B’
  80 to 100 Write ‘A’
Else
  Write ‘Invalid mark’
```

Pseudocode

```
WRITE ‘A simple program to grade a mark.’
WRITE ‘Please give the mark.’
READ the mark
CASE mark OF
  0 to 19 WRITE ‘F’
  20 to 29 WRITE ‘E’
  30 to 49 WRITE ‘D’
  50 to 64 WRITE ‘C’
  65 to 79 WRITE ‘B’
  80 to 100 WRITE ‘A’
ELSE
  WRITE ‘Invalid mark’
```

Pascal program

```
PROGRAM gradeAtoF;
VAR
mark : INTEGER;
BEGIN
   WRITELN ('A simple program to grade a mark');
   WRITELN ('Please give the mark.');
   READ (mark);
   WRITELN (mark);
   CASE mark OF
      0..19 WRITELN ('F');
      20..29 WRITELN ('E');
      30..49 WRITELN ('D');
      50..64 WRITELN ('C');
      65..79 WRITELN ('B');
      80..100 WRITELN ('A');
   ELSE
      WRITELN ('Invalid mark');
   END;
  WRITELN ('Press Enter to finish.');
   READLN;
END.
```

Python program

Python does not have a case statement. Us multiple IF statements.
mark = 0
print ('A simple program to grade a mark.')
mark = input('Please give the mark ')
if int(mark) >= 0 and int(mark) < 20:
  print('F')
if int(mark) >= 20 and int(mark) < 30:
  print('E')
if int(mark) >= 30 and int(mark) < 50:
  print('D')
if int(mark) >= 50 and int(mark) < 65:
  print('C')
if int(mark) >= 65 and int(mark) < 80:
  print('B')
if int(mark) >= 80:
  print('A')

User documentation:

This program determines a student's grade depending on the mark scored. The input is the mark scored.
Limitations: The input must be a number.

Test data:
This is a simple program so test data will include a few examples of expected input values and a few examples of more extreme values. Test data values that do not meet this criteria the expected result will be a run-time error.

Input values	Reason	Expected Result	Actual result
55	typical values	C	C
75	typical values	B	B
0	extreme values	F	F
1000000	extreme values	Invalid mark	Invalid mark
two	invalid type	error	error 106

Caption: Test data for testing a simple grading program A to F.