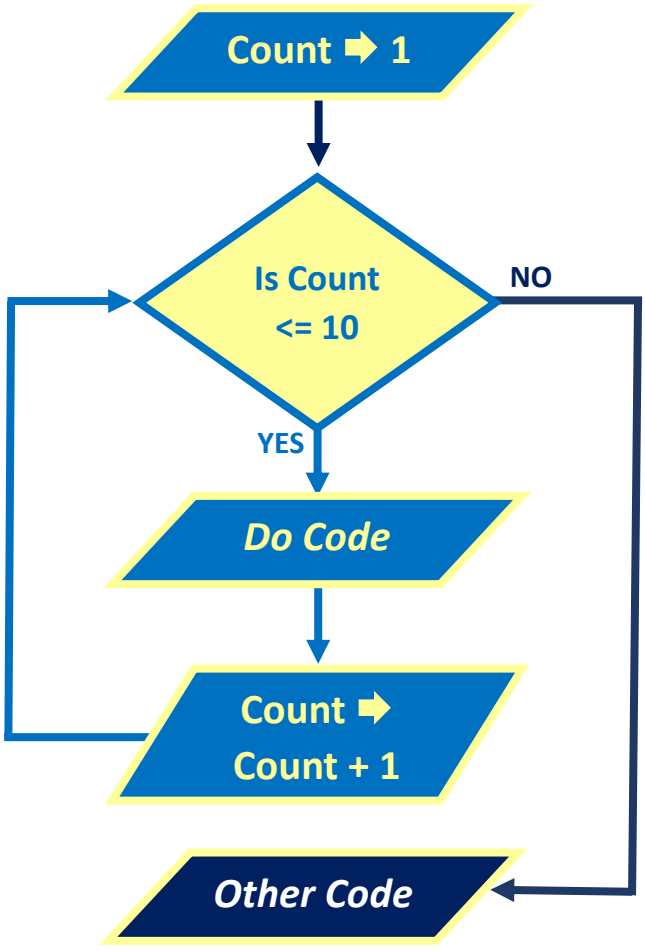
# Mr Long Summary ON FOR LOOPS

## Iteration Programming

- THREE ways the code is executed:
  - Sequential – when each line is executed in order from first line to the last line.
  - Selection – when you select which code will be executed based on a condition.
  - *Iteration – repeatedly execute code for a specific number of times.*

- Example of Iteration using a flowchart:



# FOR Loop Statement

- A **for** loop is used when you want to *repeatedly* execute a section of code a specified number of times.
  In other words: If you know how many times to repeat a section of code, use a for loop.
  ***NOTE:*** *The number of times the loop must be repeated will be known before the loop starts.*

**General Structure – FOR Loop Statement**

```
for <loop var> := <start> to <end> do
  begin
    <statement> ;
    ....
    <statement> ;
  end ;
```

  - ***loop var***(iable) is a counter that tracks how many times the loop has been executed. It must be ordinal (integer or char).
  - The loop variable starts at the value assigned at the ***start***.
  - Once the loop variable is greater than the ***end*** value, the loop will stop.
  - The loop variable will increment by one each time the loop is executed.
- The start and end values may be values or variables, if they are of the same data type of the loop variable.
  **HINT:** *Although you don't need the* ***begin end*** *if only executing one statement, I suggest always using them so that you don't get confused if you decide to add more statements to the FOR loop statement later.*
  ***WARNING:*** *Do NOT place a semi-colon (;) after the do operator as then the for loop will "do nothing" a certain number of times.*

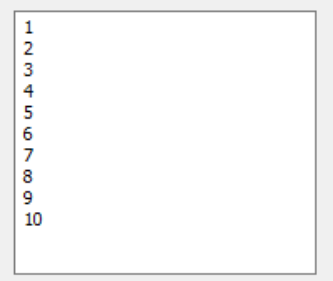
**Example 1**

```
for i := 1 to 10 then
 begin
   memDisplay.lines.add( 'Hello' ) ;
end ;
```


***Explanation:***
The code example will display the word "*Hello*" in the ***memDisplay*** memo control, 10 times.

**Example 2**

```
for i := 1 to 10 then
 begin
   memDisplay.lines.add( IntToStr( i ) ) ;
end ;
```

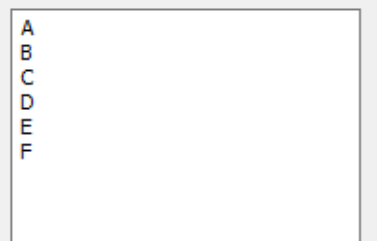



***Explanation:***
The code example will display the looping variable i in the ***memDisplay*** memo control, 10 times. It will display a 1, 2, 3, 4 until 10, each on a new line.

A char data type (character) can also be used for the loop variable. In the example below, c is of type char.

**Example 3**

```
for c := 'A' to 'F' then
 begin
   memDisplay.lines.add( c ) ;
end ;
```




***Explanation:***
The code example will display the looping variable c in the ***memDisplay*** memo control, 6 times. It will display an A, B, C until F, each on a new line.

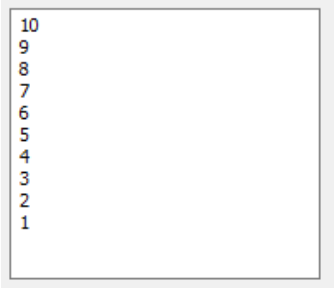
***WARNING:*** *Don't use uppercase and lowercase characters in the same for loop.*

## Loop in REVERSE order

- If you want the looping variable to DECREASE BY 1 instead of increasing, then:
  - Replace the ***TO*** operator with ***DOWNTO***
  - Start value must be a greater number than the end number
    Example: ***for*** *i* := ***10*** ***downto*** ***1*** ***do***

**Example 4**

```
for i := 10 downto 1 then
 begin
   memDisplay.lines.add( IntToStr( i ) ) ;
end ;
```

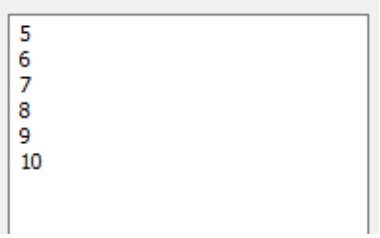



***Explanation:***
The code example will display the looping variable i in the ***memDisplay*** memo control, 10 times, in reverse order. It will display a 10, 9, 8, 7 until 1, each on a new line.

The start and end values may be values or variables, if they are of the same data type of the loop variable.

**Example 5**

```
iStart := 5 ;
iEnd := 10
  for i := iStart to iEnd then
   begin
     memDisplay.lines.add( IntToStr( i ) ) ;
  end ;
```




***Explanation:***
The code example will display the looping variable i in the ***memDisplay*** memo control, starting at the value in the ***iStart*** variable, until the value in the ***iEnd*** variable, each on a new line.

## Using calculations inside a for loop

- When displaying data, you can use the #9 code for a tab and the #13 code for starting a new line:
  - Example: memDisplay( 'Hello' + **#9** + 'Goodbye' )
    *Display Hello and Goodbye on same line but separated by a tab:* *Hello* *Goodbye*
  - Example: memDisplay( 'Hello' + **#13** + 'Goodbye' )
    *Display Hello and then Goodbye on a new line:* *Hello*
    *Goodbye*

- When displaying calculations, apply the approach of calculating and displaying *ONLY* the first line of the calculation and then let the ***for loop*** repeat the process the specified number of times.
  - If the calculation involves referring to the number of times the process as been repeated, then the calculation will normally involve some interaction with the looping variable.

**Example 6**

```
for i := 1 to 10 then
 begin
   iSquare := i * i ;
   iCube := i * i * i ;
  memDisplay.lines.add( IntToStr( i ) + #9 + IntToStr( iSquare ) + #9 + IntToStr( iCube ) ) ;
end ;
```

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 2 | 4 | 8 |
| 3 | 9 | 27 |
| 4 | 16 | 64 |
| 5 | 25 | 125 |
| 6 | 36 | 216 |
| 7 | 49 | 343 |
| 8 | 64 | 512 |
| 9 | 81 | 729 |
| 10 | 100 | 1000 |

***Explanation:***

The code example will display a list of numbers from 1 to 10, with their corresponding squared ( *number²* ) and cubed ( *number³* ) values in the ***memDisplay*** memo control.

- In the example above, each column represents a different value or calculation:
  - Column 1 wants to display a value from 1 to 10, which is the same as the looping variable.
  - Column 2 is a value that is the square of the value in column 1, which is the looping variable.
  - Column 3 is a value that is the cube of the value in column 1, which is the looping variable.
- Try finding the relationship between the calculation you are wanting to repeat with a value that is looping from 1, 2, 3, 4, etc (*which is your looping variable*)

## Calculating a sum or counting values

- If you require to count or total (sum) values inside a loop, there are two steps involved:
  - Initialise the variable used to store the count or sum. Give it a starting value. This is done BEFORE the loop.

    Example: *iSum := 0 ;*

  - Inside the loop you apply the change to the count or sum variable so that it takes the variable's current value, makes the change and stores the new value back into the count or sum variable.

    Example of summing:

| **iSum** | := | **iSum** | + | *iChange* |
|---|---|---|---|---|
| Store answer as the ***NEW*** *iSum* value | | Take the ***CURRENT*** value of *iSum* | | Apply the change to *iSum* |

Example of counting:

| iCount | := | iCount | + | 1 |
|---|---|---|---|---|
| Store answer as the ***NEW*** *iCount* value | | Take the ***CURRENT*** value of *iCount* | | Apply the change to *iCount* |

*This can also be written as:* ***Inc( iCount )*** *;*

**Example 7**

```
iSum := 0            //initialise iSum
 for i := 1 to 10 then
  begin
    iSum := iSum + i ;
  end ; //end of for loop
showmessage( IntToStr( iSum ) ) ;
```

***Explanation:***
The code example sums (adds up) all the values from 1 to 10 and displays the answer after the loop in a ***showmessage*** procedure. (*iSum = 55*)

**Example 8**

```
iCount := 0              //initialise iCount
 for i := 1 to 100 then
  begin
   if i MOD 4 = 0 then     //if loop variable is divisible by 4 with no remainder
    begin
     inc( iCount ) ;
    end ; //end of if
  end ; //end of for loop
showmessage( IntToStr( iCount ) ) ;
```

***Explanation:***
The code example counts how many values from 1 to 100 that are divisible by 4 (with no remainder) and displays the answer after the loop in a ***showmessage*** procedure. (*iCount = 25*)

---

**Additional Links:**

- Youtube video playlist:
  https://www.youtube.com/watch?v=Gx7lUiWF0pI&list=PLxAS51iVMjv8PgHNprN6kISTSUgOQ4AhD
- Google drive resource activities:
  https://tinyurl.com/MLE-G10IT-SelectionProgramming