Subject: Re: How to change the increment of FOR loop?
Posted by Allan Whiteford on Fri, 13 Mar 2009 13:44:55 GMT
View Forum Message <> Reply to Message

>
> On Thu, 12 Mar 2009, Allan Whiteford wrote:
>

>>
>>>
>>> On Wed, 11 Mar 2009, Hu wrote:
>>>
>>>> Hi, there
>>>> I got a question relating to the variational increment of FOR
>>>> Statement. I used the code like this:
>>>> ; ;; begin
>>>> increment =20
>>>> FOR j = 0, N-1, increment DO BEGIN
>>>> IF a gt 0 THEN BEGIN
>>>> ;.....
>>>> increment=10
>>>> ENDIF ELSE BEGIN
>>>> ;.....
>>>> increment=12
>>>> ENDELSE
>>>> ENDFOR
>>>> ;; end
>>>> > But the increment have not changed according to IF... ELSE ...
>>>> statement..
>>>> What I want is to change the increment according to different
>>>> conditions. that is, if variable a is greater than 0, the
>>> increment of
>>>> the FOR loop will be 10 , not its default value 20.....
>>>> > Is there any solution to this problem?
>>>
>>>
>>> increment is evaluated only once, before the loop starts (and it is
>>> copied
>>> to a temporary variable you can not reach). You have to create a loop
>>> manually:
>>>
>>> j=0
>>> increment=20
>>> while 1 do begin
>>>
>>> IF a gt 0 THEN BEGIN
>>> ;.....

```
>>>          increment=10
>>>       ENDIF ELSE BEGIN
>>>          ;......
>>>          increment=12
>>>       ENDELSE
>>>
>>>   j=j+increment
>>>   if j ge n then break
>>>
>>>   endwhile
>>>
>>>
>>>   regards,
>>>   lajos
>>
>>
>>  Although probably something like:
>>
>>  j=0
>>  increment=20
>>  while ((j+=increment) lt n) do begin
>>
>>     IF a gt 0 THEN BEGIN
>>          ;......
>>          increment=10
>>     ENDIF ELSE BEGIN
>>          ;......
>>          increment=12
>>      ENDELSE
>>
>>  endwhile
>>
>>  is easier to recognise as a "loop" from the point of view of
>>  understanding the code later for maintenance.
>
>
>  It is not the same loop :-) j starts from increment insted of 0. In this
>  form you should use j=-increment.
>
>  regards,
>  lajos
```

Pah, details - so long as the step size is small enough the start point hardly ever matters :).

Seriously, though, good point - well made.

Corrected code posted here for future reference:

```
increment=20
j=-increment
while ((j+=increment) lt n) do begin

   IF a gt 0 THEN BEGIN
      ;......
      increment=10
   ENDIF ELSE BEGIN
      ;......
      increment=12
    ENDELSE

endwhile
```

although it doesn't look nearly as nice as it did I still think a loop should look like a loop to make it easier to parse with your eyes.

I guess one could use repeat...until to get rid of the above issue but that's a bit ugly - I prefer the loop construct to appear at the top.

Thanks,

Allan
