

Olimpiada estatal de informática

Sesiones en línea

Martin Ibarra Romero

Instrucciones

Sucedede(n) Regresa el valor de $n + 1$ (Incrementa)

Precede(n) Regresa el valor de $n - 1$ (Decremento)

Si-es-cero(n) Regresa verdadero si n es cero o falso en otro caso

Por ser 3 funciones se evalúan como funciones (similares a las de excel o de una calculadora)

Notas de funciones

Las funciones se evalúan de forma similar a cuando usamos una calculadora o a una formula de excel.

Esto significa que el valor de la variable nunca es modificado sino que lo que se modifica es el valor que regresa.

Recuerde que los paréntesis se resuelven de adentro hacia afuera de manera similar al algebra.

Sucede Suponga N=5

```
10 (* si n vale 5*)
11 repetir sucede(sucede(Sucede(n))) veces inicio
12     avanza;
13 fin;
```

Al momento de ejecutarse la línea 11 con el valor de n=5 sucede lo siguiente:

Línea a ejecutar: repetir sucede(sucede(Sucede(n)))

Paso 1: se sustituye la n por el valor por 5 repetir:

repetir sucede(sucede(Sucede(5))) veces inicio

Paso 2: se elimina el primer sucede y el 5 se incrementa en 1:

repetir sucede(sucede(6)) veces inicio

Paso 3: se elimina el segundo sucede y el 6 se incrementa en 1:

repetir sucede(7) veces inicio

Paso 4: se elimina el tercer sucede y el 7 se incrementa en 1:

repetir 8 veces inicio

Paso 5: Se ejecuta la línea 11 y el repetir se realiza 8 veces

Como se puede apreciar el valor de n no se modifica sin embargo el repetir se realiza 8 veces

Precede Suponga N=11

```
10 (* si n vale 11 *)  
11 repetir precede(precede(n)) veces inicio  
12     avanza;  
13 fin;
```

Al momento de ejecutarse la línea 11 con el valor de n=11 sucede lo siguiente:

Línea a ejecutar: repetir precede(precede(n))

Paso 1: se sustituye la n por el valor por 11 repetir:

repetir precede(precede(11)) veces inicio

Paso 2: se elimina el primer precede y el 10 disminuye en 1:

repetir precede(10) veces inicio

Paso 3: se elimina el segundo precede y el 11 disminuye en 1:

repetir 9 veces inicio

Paso 4: Se ejecuta la línea 11 y el repetir se realiza 9 veces

Como se puede apreciar el valor de n no se modifica, sin embargo el repetir se realiza 9 veces

Verificar un valor

Para verificar si el valor de n es igual a 1, 2, 3, 4, etc. lo que se debe hacer es antecederle tantos precedes como valor queremos verificar y preguntar si es cero, por ejemplo suponga una $n=2$ y queremos saber si es 2, y en caso de $n=2$ sea cierto que karel avance.

10 (* si n vale 2 *)

11 Si si-es-cero(precede(precede(n))) entonces inicio

12 avanza;

13 fin;

Al momento de ejecutarse la línea 11 con el valor de $n=2$ sucede lo siguiente:

Línea a ejecutar: Si si-es-cero(precede(precede(n))) entonces inicio

Paso 1: se sustituye la n por el valor por 2 repetir:

Si si-es-cero(precede(precede(2))) entonces inicio

Paso 2: se elimina el primer precede y el 2 disminuye en 1:

Si si-es-cero(precede(1)) entonces inicio

Paso 3: se elimina el segundo precede y el 1 disminuye en 1:

Si si-es-cero(0) entonces inicio

Paso 4: El si-es-cero devuelve verdadero y se ejecuta el entonces

Como se puede apreciar el valor de n no se modifica, sin embargo se ejecuta el entonces del si, ósea se cumple $n=2$ y se realiza el avanza.

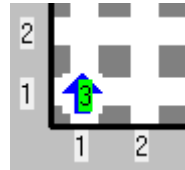
Como podemos tomar un valor del mundo de karel

La manera de tomar un valor en el mundo de karel es realizar un procedimiento recursivo y por cada llamado recursivo realizar un sucede. Cuando la condición de continuar los llamados recursivos termine, nosotros tendremos el valor, es necesario llamar a nuestro procedimiento recursivo con un valor de 0 la primera vez.

Nota: no olvide que por ser un procedimiento recursivo se ejecutan los retornos de recursividad y es responsabilidad de usted decidir si los va utilizar o no dentro de una solución

Ejemplo

Veamos el siguiente ejemplo, karel se encuentra parado sobre un montón de zumbadores y lo que se desea es tomarlo en un parámetro y avanzar hacia adelante por medio del parámetro.

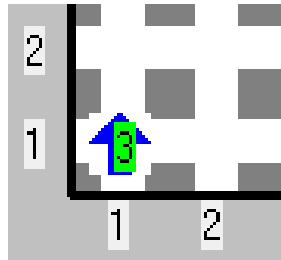


```
define-nueva-instruccion contar(n) como inicio
  si junto-a-zumbador entonces inicio
    coge-zumbador;
    contar(sucede(n));
  fin sino inicio
    (* Aquí tengo el lugar*)
    repetir n veces inicio
      avanza;
    fin;
  fin;
fin;
```

```
inicia-ejecucion
  contar(0);
  apagate;
termina-ejecucion
```


Ejecución 1

Veamos el siguiente ejemplo, karel se encuentra parado sobre un montón de zumbadores y lo que se desea es tomarlo en un parámetro y avanzar hacia adelante por medio del parámetro.



```
inicia-ejecucion
  contar(0);
  apagate;
termina-ejecucion
```

```
define-nueva-instruccion contar(n) como inicio
  si junto-a-zumbador entonces inicio
    coge-zumbador;
    contar(sucede(n));
  fin sino inicio
    repetir n veces inicio
      avanza;
    fin;
  fin;
fin;
```

Note que contar se ejecuta la primera vez con cero tomará un zumbador y como sucede de cero es 1, contar se ejecutará con 1

Por partes 1

```
define-nueva-instruccion contar(n=0) como
```

```
inicio
```

```
  si junto-a-zumbador
```

```
    coge-zumbador
```

```
    contar(sucede)
```

```
  fin sino inicio
```

```
    repetir n veces
```

```
      avanza;
```

```
    fin;
```

```
  fin;
```

```
fin;
```

```
define-nueva-instruccion contar(n=1) como
```

```
inicio
```

```
  si junto-a-zumbador entonces inicio
```

```
    coge-zumbador;
```

```
    contar(sucede(n=1));
```

```
  fin sino inicio
```

```
    repetir n veces inicio
```

```
      avanza;
```

```
    fin;
```

```
  fin;
```

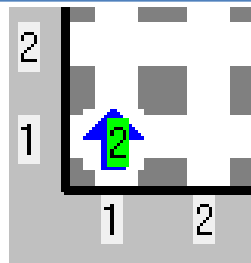
```
fin;
```

```
inicia-ejecucion
```

```
  contar(0);
```

```
  apagate;
```

```
termina-ejecucion
```

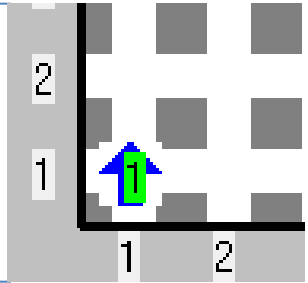


Se toma un zumbador y como sucede de 1 es 2, se ejecutará contar con 2

Por partes 2

```
define-nueva-instruccion contar(n=0) como
inicio
  define-nueva-instruccion contar(n=1) como
  si junto-a-zumbador entonces inicio
    co
    co
    fin s
    re
    a
    fin
  fin;
fin;
fin;
```

```
inicia-ejecucion
  contar(0);
  apagate;
termina-ejecucion
```



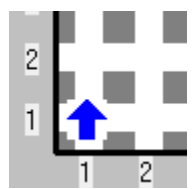
Se toma un zumbador y como sucede de 2 es 3, se ejecutará contar con 3

Por partes 3

```
define-nueva-instruccion contar(n=0) como
inicio
  define-nueva-instruccion contar(n=1) como
  si junto-a-zumbador entonces inicio
  coge-zumbador;
  contar(sucede(n=3));
  fin sino inicio
  repetir n=3 veces inicio
  avanza;
  fin;
fin;
fin;
```

Como no hay zumbadores se termina la condición de recursividad y se ejecuta el repetir 3 veces.

```
inicia-ejecucion
  contar(0);
  apagate;
termina-ejecucion
```



Note que en la pila se quedan los regresos recursivos