Ejercicio TC32-39p

En varias ramas de la Física, especialmente en Óptica, aparece la necesidad de calcular repetidamente la función conocida como *Seno Integral*, $Si(x)$, definida como

$$Si(x) = \int_0^x \frac{\mathrm{sen}(t)}{t} dt = \sum_{n=0}^{\infty} \frac{(-1)^n \cdot x^{2n+1}}{(2n+1) \cdot (2n+1)!}, \quad x > 0.$$

Hay que escribir una rutina, en lenguaje ensamblador del 80x86/x87, que calcule la función *Seno Integral* según el desarrollo en serie indicado. A la rutina se le deben pasar por la pila los siguientes parámetros:

- argumento de la función, x,
- error máximo admisible en el cálculo de la función, err,
- número máximo de iteraciones permitidas, N,

de la forma

$$\text{SenoIntegral}(x, err, N)$$

siguiendo el convenio del lenguaje C de paso de parámetros, primero el de más a la derecha y se termina con el de más a la izquierda. Por ejemplo, se llamaría a la rutina así:

```
push    N
push    err
push    x
call    SenoIntegral
```

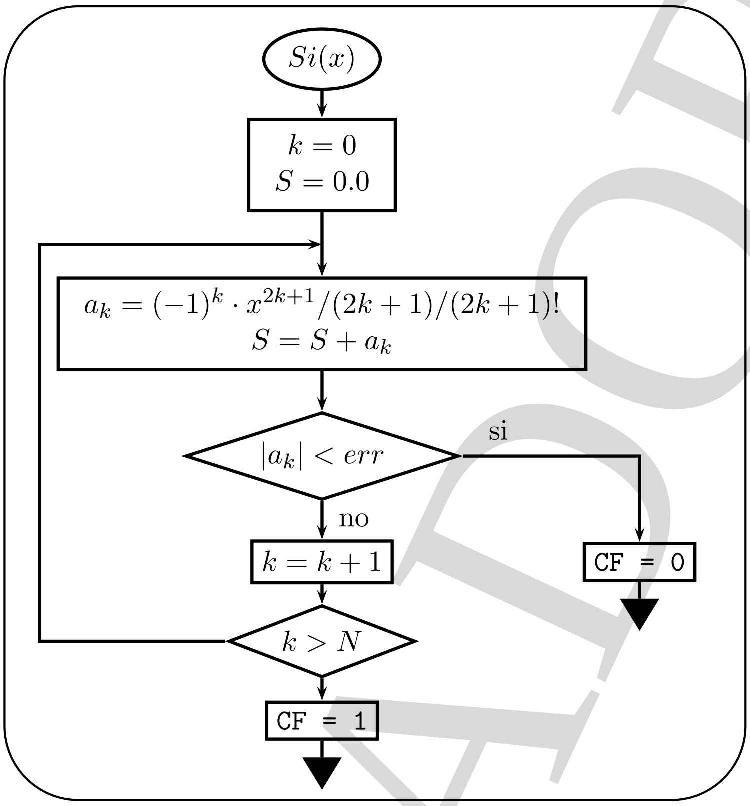
Si se logra calcular la función en menos de N iteraciones con error menor que err el resultado se devuelve en el registro `ST(0)` de la FPU y el número de iteraciones realizadas en `AX`, poniendo `CF`=0. En caso de agotar el número de iteraciones N sin lograr un error menor de err se devuelve la aproximación obtenida en `ST(0)` y el indicador de carry, `CF`=1, para señalar la no convergencia.

Solución

La serie

$$Si(x) = \sum_{n=0}^{\infty} \frac{(-1)^n \cdot x^{2n+1}}{(2n+1) \cdot (2n+1)!} = x - \frac{x^3}{3 \cdot 3!} + \frac{x^5}{5 \cdot 5!} - \frac{x^7}{7 \cdot 7!} + \dots$$

se calculará iterativamente, sumando/restando un término en cada iteracción. Un primer esquema de cálculo sería el que se plantea en el siguiente diagrama de flujo:



ya que, al ser una serie alterna, el error es menor que el nuevo término que se añade:

$$err < \epsilon_k = |S_k - S_{k-1}| = |a_k|$$

Los términos de la serie pueden calcularse recurrentemente:

$$\frac{a_{k+1}}{a_k} = \frac{\frac{(-1)^{k+1} \cdot x^{2k+3}}{(2k+3)\cdot(2k+3)!}}{\frac{(-1)^k \cdot x^{2k+1}}{(2k+1)\cdot(2k+1)!}} = -\frac{(2k+1)\cdot x^2}{(2k+3)(2k+2)(2k+3)}$$

O sea:

$$\boxed{a_{k+1} = -\frac{(2k+1)\cdot x^2}{(2k+2)\cdot(2k+3)^2} \cdot a_k} \qquad k = 0, 1, \ldots$$

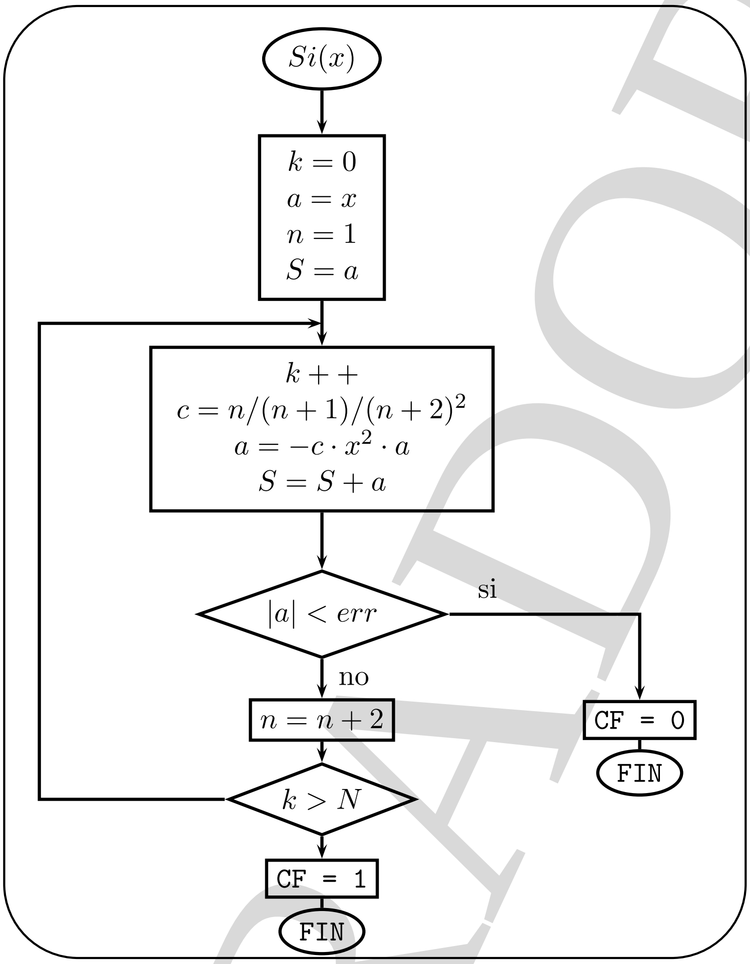
Los coeficientes son:

$$\frac{(2k+1)}{(2k+2)(2k+3)^2}$$

$\frac{1}{2\cdot 3^2}$, $\frac{3}{4\cdot 5^2}$, $\frac{5}{6\cdot 7^2}$, … que pueden irse calculando por su cuenta así: $n_0 = 1$, $n_{k+1} = n_k + 2$,

$$c_k = \frac{n_k}{(n_k+1)\cdot(n_k+2)^2} \; .$$

Poniendo todo esto junto:



Se precisan dos contadores enteros, k y n. El primero no participa de los cálculos y se puede mantener en un registro. El segundo, n, hay que pasárselo a la FPU y conviene mantenerlo en memoria, mejor en memoria local de la rutina.

Los valores en coma flotante que hay que mantener de un ciclo al siguiente son: el parámetro de entrada x, el término a y la suma parcial S. Los tres valores pueden mantenerse en los registros de la FPU. El coeficiente c se calcula en cada iteración.

Conviene que el orden de los valores en los registros de la FPU sea

a	st(0)
x	st(1)
S	st(2)
…	

al comenzar el bucle de la iteración, evolucionando así el contenido de los registros de la FPU:

st(0)	st(1)	st(2)	st(3)	-
n	a	x	S	
$n/(n+1)$	a	x	S	
$a \cdot n/(n+1)$	x	S		
x	$a \cdot n/(n+1)$	x	S	
$x/(n+2)$	$a \cdot n/(n+1)$	x	S	
$x/(n+2)$	$(x/(n+2)) \cdot (a \cdot n/(n+1))$	x	S	
$a = (x/(n+2)) \cdot (x/(n+2)) \cdot (a \cdot n/(n+1))$	x	S		
$a = -a$	x	S		
a	x	$S = S + a$		
a	a	x	S	
$\|a\|$	a	x	S	

y se compara $|a|$ con err, quitando a $|a|$ y quedando la pila como al principio.

El código de la rutina puede ser el siguiente:

```
SenoIntegral    proc near
        push    bp
        mov     bp,sp
        sub     sp,2        ; hace sitio para n en la pila
        ;   ***
        fld     DWORD PTR [bp+4]  ; x |
        xor     ax,ax       ; k = 0
        mov     WORD PTR [bp-2],1 ; n = 1
        fld     st(0)             ; x | S = x |
        fld     st(0)             ; a = x | x | S |
@bucle:
        inc     ax
        fild    WORD PTR [bp-2]   ; n | a | x | S |
        inc     WORD PTR [bp-2]   ; n+1
        fidiv   WORD PTR [bp-2]   ; n/(n+1) | a | x | S |
        inc     WORD PTR [bp-2]   ; n+2
        fmul                      ; a·n/(n+1) | x | S |
        fld     st(1)             ; x | a·n/(n+1) | x | S |
        fidiv   WORD PTR [bp-2]   ; x/(n+2) | a·n/(n+1) | x | S |
        fmul    st(1),st(0)
        ; x/(n+2) | (x/(n+2))·(a·n/(n+1)) | x | S |
        fmul                      ; a' | x | S |
        fchs                      ; a | x | S |
        fadd    st(2),st(0)       ; a | x | S=S+a |
        fld     st(0)             ; a | a | x | S |
        fabs                      ; abs(a) | a | x | S |
        fcomp   DWORD PTR [bp+8]  ; ¿abs(a) < err?
        push    ax
        fstsw   ax
        sahf
        pop     ax
        jb      @converge
        cmp     ax,WORD PTR [bp+12] ; ¿k > N?
```

```
        ja      @no_converge
        jmp     @bucle
@converge:
        clc                             ; CF = 0
        jmp     @final
@no_converge:
        stc                             ; CF = 1
@final:
        fstp    st(0)
        fstp    st(0)
        mov     sp,bp
        pop     bp
        ret
SenoIntegral    endp
```

La pila en el punto `***` del programa es:

	libre	
`SP >`	`n`	-2
	`BP antiguo`	`< BP`
	`direc. vuelta`	$+2$
	`x`	$+4$
	`err`	$+8$
	`N`	$+12$