PHOENIX LANGUAGE
HASH EXTENSION OF LASER GENIUS

(FRAME POINTER HANDLING)

Example1:

Laser Genius Phoenix source	Laser Genius Phoenix output

`#STACK`	`ld    ix,0`
	`add   ix,sp`
	`dec   sp`

`jp    start`	`jp    start`

`fce:   #FNC    INT`
`xx:    #PRM    INT`
`yy:    #PRM    INT`
 	
`#BEGIN`
`#DUE   xx+yy`
 	
`#END`
	`fce:   push   de`
	`push   ix`
	`ld     ix,0`
	`add    ix,sp`
	`dec    sp`
	`ld     l,(ix+6)`
	`ld     h,(ix+7)`
	`push   hl`
	`ld     l,(ix+8)`
	`ld     h,(ix+9)`
	`push   hl`
	`call   __CALLEE_PH__intrin_uadd16`
	`pop    hl`
	`ld     sp,ix`
	`pop    ix`
	`pop    de`
	`ret`

`start:  #DUE    fce(1,2)`	`start:  ld     hl,1`
	`push   hl`
	`ld     hl,2`
	`push   hl`
	`ld     de,4`
	`call   fce`
	`ex     de,hl`
	`add    hl,sp`
	`ld     sp,hl`
	`ex     de,hl`

`ret`	`ret`

#LIB	```
	__CALLEE_PH__intrin_uadd16:
	        pop hl
	        pop de
	        ex (sp),hl
	        add hl,de
	        jr nc,l_65dc
	        set 0,(ix-1)
	l_65dc: ex (sp),hl
	        jp (hl)
	``

Example2:

Laser Genius Phoenix source	Laser Genius Phoenix output

`#STACK`	`ld      ix,0`
	`add     ix,sp`
	`dec     sp`

`        jp      start`	`        jp      start`

`fce:    #FNC    INT`
 	
`xx:     #PRM    INT`
`yy:     #PRM    INT`
`local:  #DS     INT,10`
 	
`        #BEGIN`
`        #DUE    xx*yy`
 	
`        #END`
	`fce:    push    de`
	`        push    ix`
	`        ld      ix,0`
	`        add     ix,sp`
	`        ld      hl,-(10*sizeof(INT) + 1)`
	`        add     hl,sp`
	`        ld      sp,hl`
	`        ld      l,(ix+6)`
	`        ld      h,(ix+7)`
	`        push    hl`
	`        ld      l,(ix+8)`
	`        ld      h,(ix+9)`
	`        push    hl`
	`        call    __CALLEE_PH__intrin_umult16`
	`        pop     hl`
	`        ld      sp,ix`
	`        pop     ix`
	`        pop     de`
	`        ret`

`start:  #DUE    fce(1,2)`	`start:  ld      hl,1`
	`        push    hl`
	`        ld      hl,2`
	`        push    hl`
	`        ld      de,4`
	`        call    fce`
	`        ex      de,hl`
	`        add     hl,sp`
	`        ld      sp,hl`
	`        ex      de,hl`

`        ret`	`        ret`

#LIB	

```
loc_umult16:   ld      c,h
               ld      a,l
               ld      b,16
               ld      hl,0
mloop:         add     hl,hl
               jr      nc,l_65e4
               set     0,(ix-1)
l_65e4:        rla
               rl      c
               jr      nc,l_65f0
               add     hl,de
               jr      nc,l_65f0
               set     0,(ix-1)
l_65f0:        djnz    mloop
               ret
__CALLEE_PH__intrin_umult16:
               pop     hl
               pop     de
               ex      (sp),hl
               call    loc_umult16
               ex      (sp),hl
               jp      (hl)
```