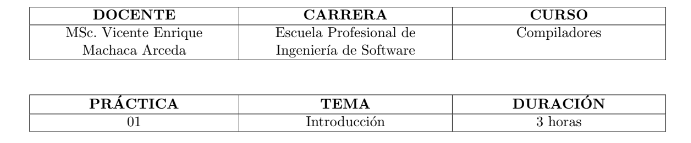
**Practica 1**



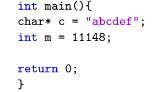
Integrantes:

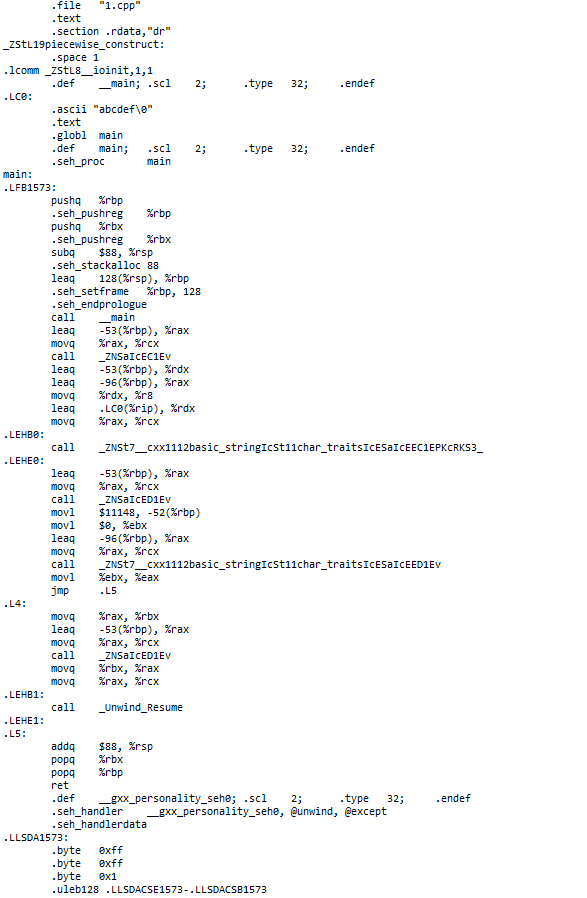
Carlos Corrales

Leonardo Deza

1. **Ejercicios**

1.1 Redacta el siguiente código, genera el código ensamblador y explica en qué parte (del código ensamblador) se definen las variables c y m.

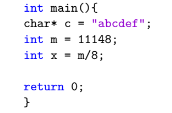




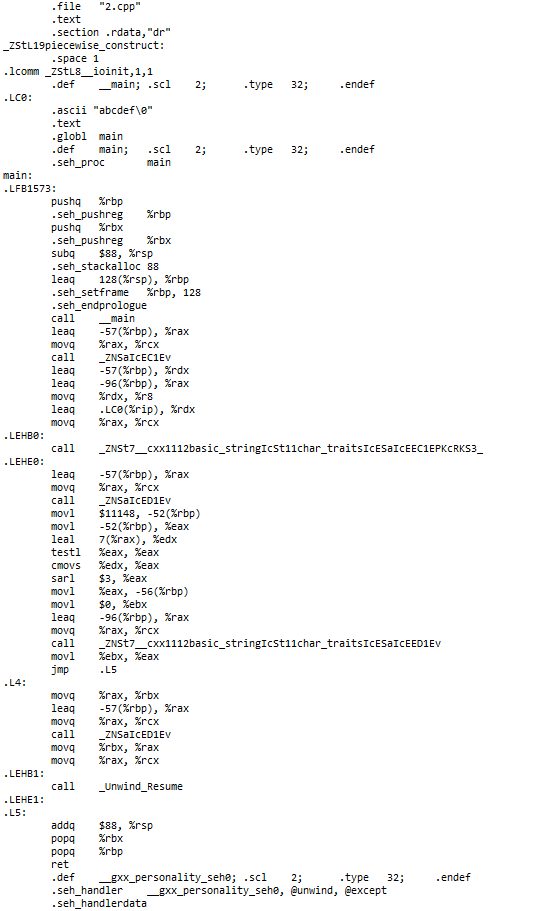
Respuesta



1.2 Redacta el siguiente código, genera el código ensamblador y explica en qué parte (del código ensamblador) se define la división entre 8.







Respuesta

movl -52(%rbp), %eax

leal 7(%rax), %edx

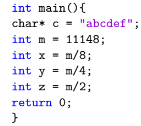
testl %eax, %eax

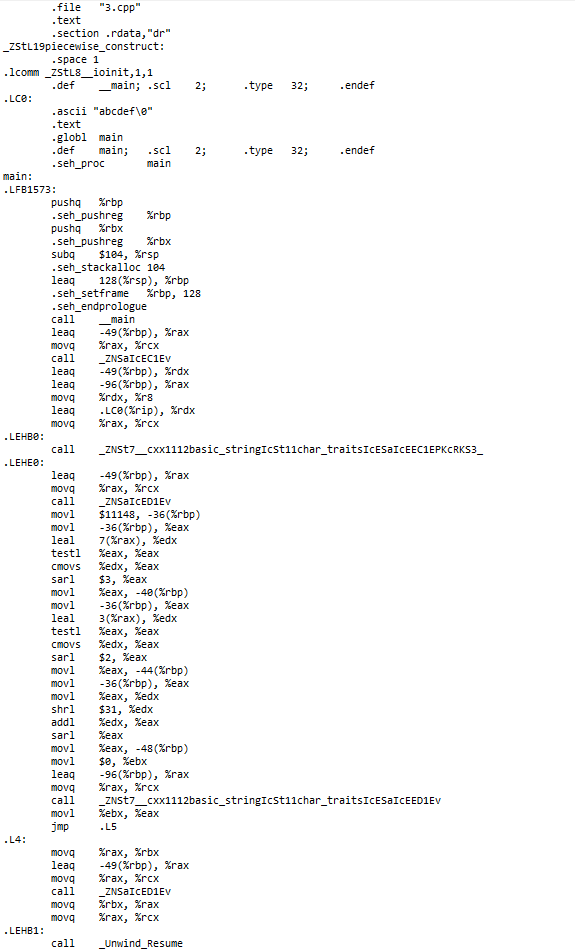
cmovs %edx, %eax

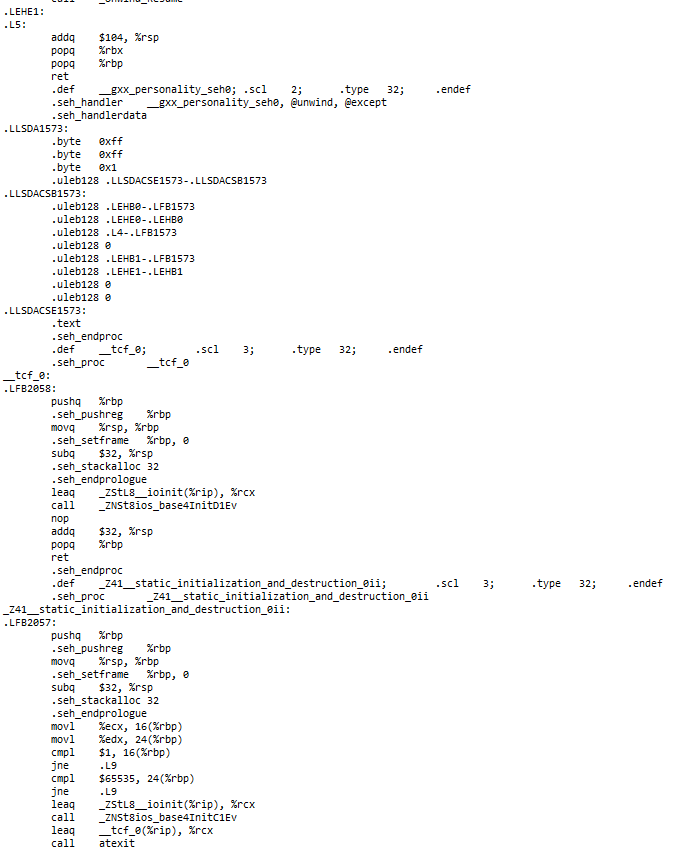
sarl $3, %eax

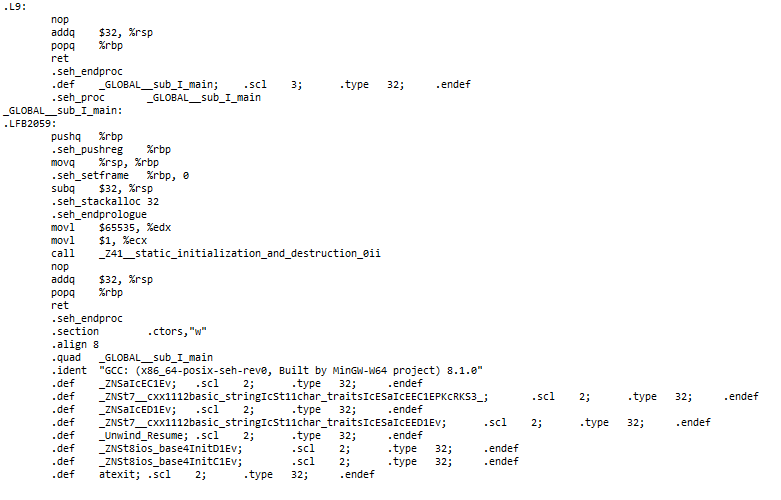
movl %eax, -56(%rbp)

1.3 Redacta el siguiente código, genera el código ensamblador y explica en qué parte (del código ensamblador) se define la división entre 4.









Respuesta

movl -36(%rbp), %eax

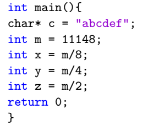
leal 3(%rax), %edx

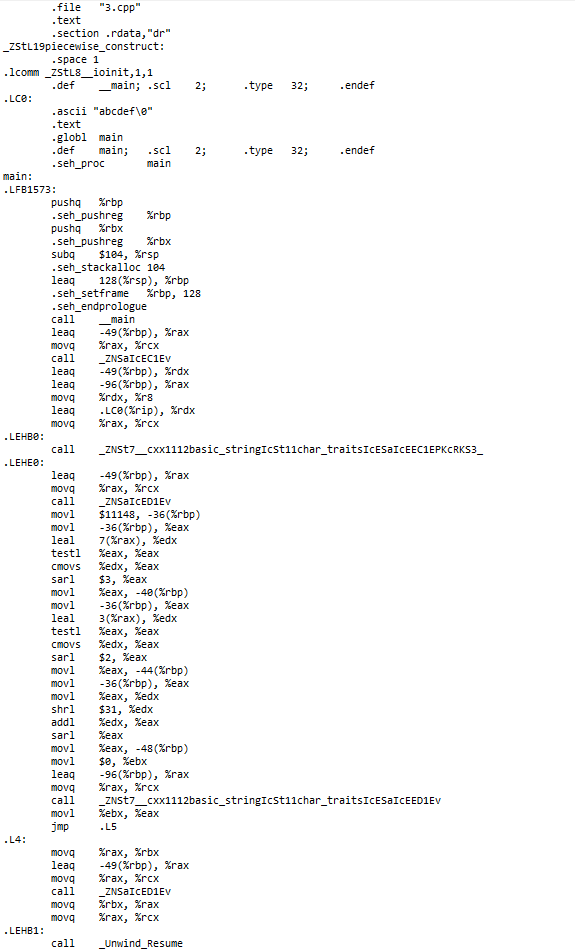
testl %eax, %eax

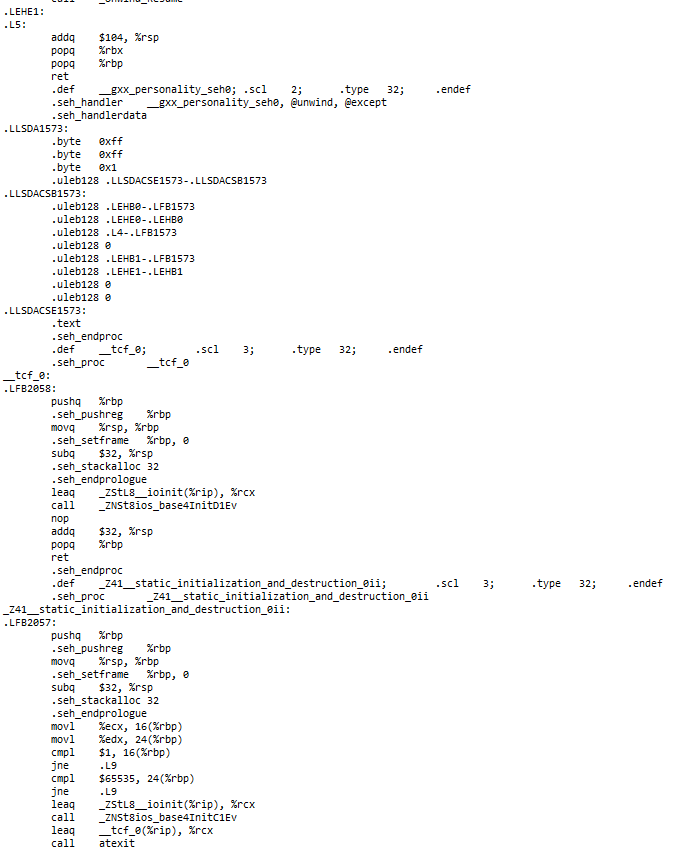
cmovs %edx, %eax

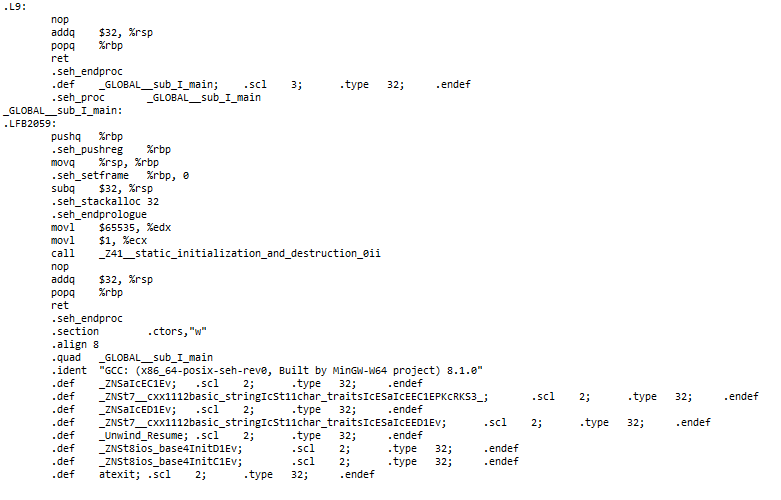
sarl $2, %eax

movl %eax, -44(%rbp)

1.4 Redacta el siguiente código, genera el código ensamblador y explica en qué parte (del códigoensamblador) se define la división entre 2.







Respuesta

movl -36(%rbp), %eax

movl %eax, %edx

shrl $31, %edx

addl %edx, %eax

sarl %eax

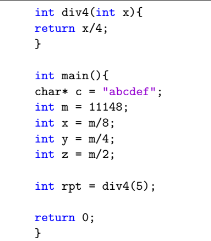
movl %eax, -48(%rbp)

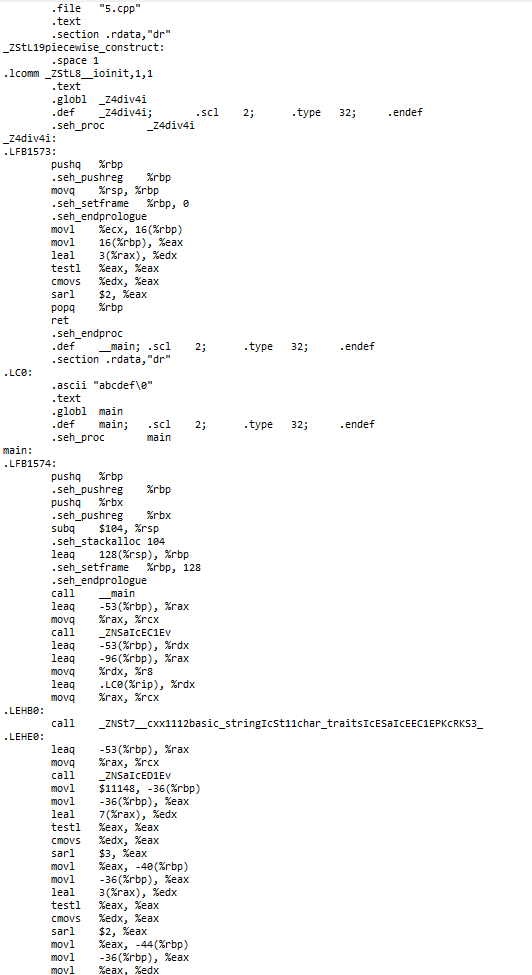
1.5 Redacta el siguiente código, genera el código ensamblador y explica:

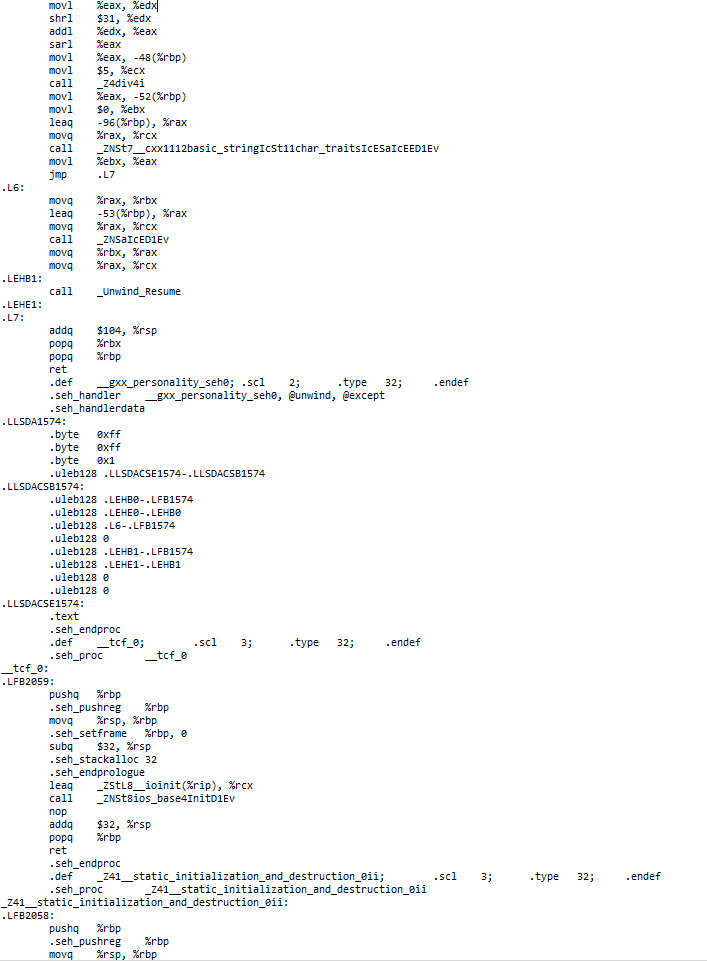
En qué parte del código ensamblador se define la función div4.

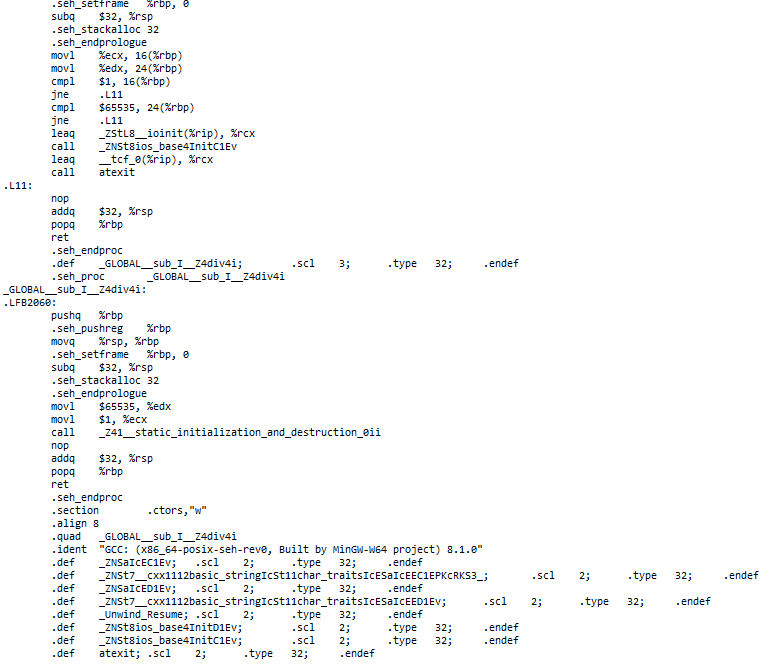
En qué parte del código ensamblador se invoca a la función div4.

En qué parte del código ensamblador dentro de la función div4 se procesa la división.









Respuesta:

1.-

lcomm ZStL8\_ioinit,1,1

.text

.globl \_Z4div4i

.def \_Z4div4i; .scl 2; .type 32; .endef

.seh\_proc \_Z4div4i

2.-

call \_Z4div4i

3.-

.LFB1573:

pushq %rbp

.seh\_pushreg %rbp

movq %rsp, %rbp

.seh\_setframe %rbp, 0

.seh\_endprologue

movl %ecx, 16(%rbp)

movl 16(%rbp), %eax

leal 3(%rax), %edx

testl %eax, %eax

cmovs %edx, %eax

sarl $2, %eax

popq %rbp

ret

.seh\_endproc

.def \_\_main; .scl 2; .type 32; .endef

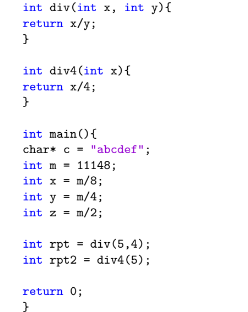
.section .rdata,"dr"

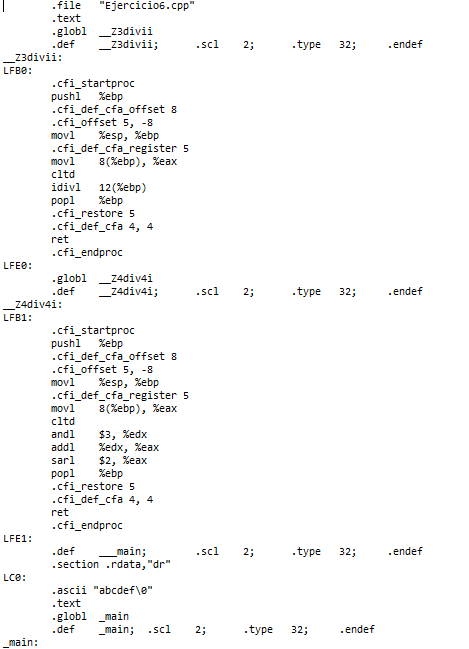
1.6 Redacta el siguiente código, genera el código ensamblador y explica:

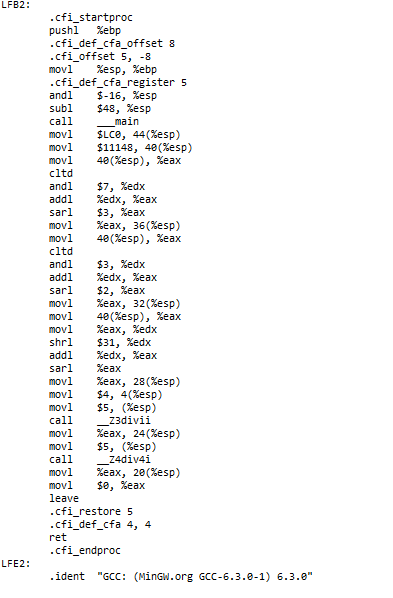
En qué parte del código ensamblador se define la función div.

En qué parte del código ensamblador se invoca a la función div.

En qué parte del código ensamblador dentro de la función div se procesa la división.







Respuesta

1.-

.globl \_\_Z3divii

.def \_\_Z3divii; .scl 2; .type 32; .endef

\_\_Z3divii:

2.-

call \_\_Z3divii

3.-

\_\_Z3divii:

LFB0:

.cfi\_startproc

pushl %ebp

.cfi\_def\_cfa\_offset 8

.cfi\_offset 5, -8

movl %esp, %ebp

.cfi\_def\_cfa\_register 5

movl 8(%ebp), %eax

cltd

idivl 12(%ebp)

popl %ebp

.cfi\_restore 5

.cfi\_def\_cfa 4, 4

ret

.cfi\_endproc

1.7 De las preguntas anteriores, se ha generado código por cada función, ambas dividen entre 4, pero difieren un poco en su implementación. Investigue a qué se debe dicha diferencia y comente cuáles podrían ser las consecuencias.

Primero existe el hecho de que además de dividir entre 4, puede dividir entre cualquier otro número, esto causa que reserve una variable extra, lo que causa una pérdida de memoria para cosas pequeñas, por otro lado, la función div4 divide únicamente entre 4, en ensamblador esta orden la toma simplemente como desplazamiento. Pero volviendo al otro caso este desplazamiento puede ser cualquier numero por lo que antes reserva una variable para esto y luego recién realiza el desplazamiento