

Nombre: andyfratelloGrupo: 88

Nombre: \_\_\_\_\_

## Hoja de respuesta al Estudio Previo

1. Explicad para qué sirven y qué operandos admiten las instrucciones:

psubb

Resta entera de bytes empaquetados.

pcmpgtb

Compara entera de bytes empaquetados determinando si el primer es más grande que el segundo.

movdqa

Mueve un double quadword alineado desde el segundo operando al primer operando.

movdqu

Mueve un double quadword no alineado desde el segundo operando al primer operando.

emms

Borra el estado de MMX. Borra un contenido por acomodar nuevo contenido.

2. La propiedad
- attribute
- y el atributo
- aligned
- sirven para:

-attribute- permite especificar atributos especiales al ser una declaración.aligned es un atributo que especifica cuántos bytes ha de estar alineada la variable.



3. Programad en ensamblador una versión de la rutina que hay en Procesar.c procurando hacerla lo más rápida posible.

<pre> movl 8(%ebp), %eax movl 12(%ebp), %ecx movl 16(%ebp), %edi movl 20(%ebp), %edx imul %edx, %edx addl %eax, %edx for: cmpl %edx, %eax jge end_for movl (%eax), %cl subl (%ecx), %cl cmpl \$0, %cl jle endif movl \$255, %cl </pre>	<pre> endif: movl %cl, (%edi) incl %eax incl %ecx incl %edi jmp end_for end_for: </pre>
--	---

4. Explicad como se puede cargar un valor inmediato en un registro xmm usando la instrucción movdqu.

Ante la instrucción `movdqu xmm1, xmm2/m128` cargar un double quadword de `xmm2/m128` a `xmm1`

no alineat

5. Programad en ensamblador una versión SIMD de la rutina que hay en Procesar.c.

<pre> movl 8(%ebp), %eax movl 12(%ebp), %ecx movl 16(%ebp), %edi movl 20(%ebp), %edx imul %edx, %edx addl %eax, %edx for: cmpl %edx, %eax jge end_for movdqu (%eax), %xmm0 movdqu (%ecx), %xmm1 pshld %xmm1, %xmm0 movdqu (%edx), %xmm1 pcmpgtd %xmm1, %xmm0 </pre>	<pre> movdqu %xmm0, (%edi) addl \$16, %eax addl \$16, %ecx addl \$16, %edi jmp for end_for: </pre>
---	--

6. Escribid un código en ensamblador que a partir de un valor almacenado en un registro averigüe si es múltiplo de 16.

<pre> andl 0x0000000F, %eax jne no_multiple multiple: no_multiple: end: </pre>	
--	--