

Nombre: Daniel Martínez Bordon Grupo: 2A

Nombre: _____

Hoja de respuesta al Estudio Previo

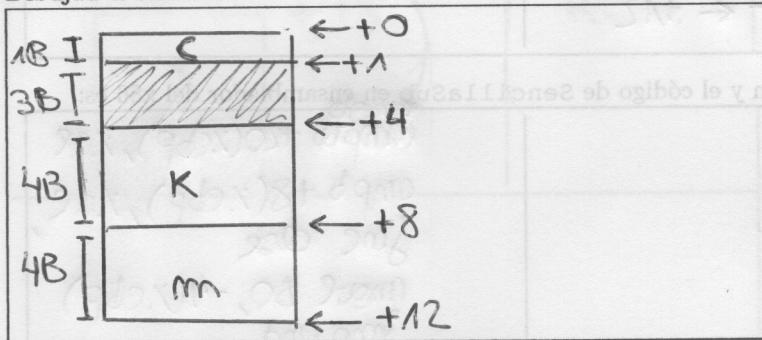
1. Explicad qué significan y para qué sirven los parámetros "argc" y "argv" del main.

argc es el nº de parámetros que recibe la rutina main (+1 ya que el nombre del programa también es un parámetro). argv es un vector de argc componentes que apunta los parámetros como strings

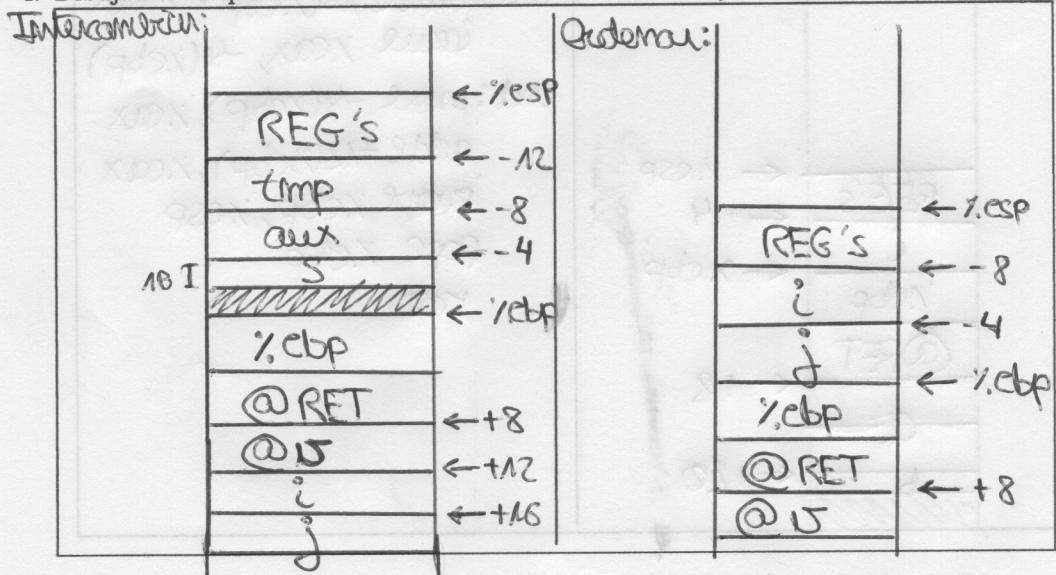
2. Explicad para qué sirve la función atoi

La función atoi convierte un string en un entero.

3. Dibujad el struct S1:



4. Dibujad el bloque de activación de las rutinas Ordenar y Intercambiar.



5. Dibujad el bloque de activación de la rutina main.

a	$\leftarrow \text{?CSP}, -412$
b	$\leftarrow -408$
m	$\leftarrow -404$
n	$\leftarrow -400$
trabat	$\leftarrow -396$
X	$\leftarrow -384$
Y	$\leftarrow -372$
Z	$\leftarrow -360$
15	
?ebp	$\leftarrow \text{?esp}$
$@RET$	$\leftarrow +8$
auge	$\leftarrow +12$
$@auge$	

Temporales:
push %ebp
movl %esp, %ebp
subl \$4, %esp

6. El bloque de activación y el código de SencillaSub en ensamblador del x86 es:

REG's	$\leftarrow \text{?esp}$
i	$\leftarrow -4$
?ebp	$\leftarrow \text{?esp}$
$@RET$	$\leftarrow +8$
a	
b	$\leftarrow +20$

```

(movb +20(%ebp), %ax
cmpb +8(%ebp), %ax
jne else
move $0, -4(%ebp)
jmp end

else: (move 16(%ebp), %eax
imul %eax, -4(%ebp))

end: move -4(%ebp), %eax
addl +12(%ebp), %eax
move %ebp, %esp
pop %ebp
ret

```

7. El bloque de activación y el código de LlamaSencilla en ensamblador del x86 es:

			LlamaSencilla:
			pushc y,ebp
			move %esp,%ebp
			pushb +8(%ebp)
			pushb 10(%ebp)
			pushb 12(%ebp)
			pushb 8(%ebp)
			call SencillaSub
			addl \$16,%esp
			move 16(%ebp),%ecx
			move (%eax,%ecx)
			move 12(%ebp),%eax
			move %ebp,%esp
			popl %ebp
			ret
REG's	← %esp		
%ebp	← %ebp		
@RET	← +8		
x			
y	← +20		