

Plancha 2 Ejercicios 6 y 7 Representación Computacional de Datos

Arquitectura del Computador

Hedman Ulises, Pitinari Tomás y Quintero Iago

6)

```
.data
format: .asciz "%ld\n"
formatHex: .asciz "%lX\n"
i: .quad 0xDEADBEEF

.text
.global main
main:
movq $format, %rdi      # El primer argumento es el formato.
movq $1234, %rsi        # El valor a imprimir.
xorq %rax, %rax         # Cantidad de valores de punto flotante.
call printf

#a
movq $format, %rdi
movq (%rsp), %rsi
xorq %rax, %rax
call printf

#b
movq $format, %rdi
movq $format, %rsi
xorq %rax, %rax
call printf

movq $formatHex, %rdi
movq $format, %rsi
xorq %rax, %rax
call printf

#El item d se puede saltar ya que es lo mismo que en el item a

movq $format, %rdi
movq 8(%rsp), %rsi
xorq %rax, %rax
call printf

movq $format, %rdi
movq (i), %rsi
xorq %rax, %rax
call printf

movq $format, %rdi
movq $i, %rsi
xorq %rax, %rax
call printf

ret
```

7)a)

```
.data
.text
.global main
main:
    movq $0xA0A0A0A0E0E0E0E0 ,%rax
    ror $32, %rax
    ret
```

b)

```
.global main
main:
    movq $0xffffffffffffffff, %rax    #numero
    xorq %rbx, %rbx    #iteraciones
    xorq %rcx, %rcx    #resultado final
    jmp while

while:
    cmpq $64, %rbx
    jz fin
    jmp cuerpo
cuerpo:
    ror $1, %rax
    incq %rbx
    adc $0, %rcx
    jmp while
fin:
    ret
```