

## Media de N enteros con signo de 32 bits en una plataforma de 32bits

### Código

```
..section .data
    .macro linea
        #.int 1,-2,1,-2
        #.int 1,2,-3,-4
        #.int 0x7fffffff,0x7fffffff,0x7fffffff,0x7fffffff
        #.int 0x80000000,0x80000000,0x80000000,0x80000000
        #.int 0xf0000000,0xe0000000,0xe0000000,0xd0000000
        #.int -1,-1,-1,-1
    .endm

    .macro linea0
        .int 0,-1,-1,-1
        #.int 0,-2,-1,-1
        #.int 16,-2,-1,-1
        #.int 60,-2,-1,-1
        #.int 90,-2,-1,-1
        #.int -12,-2,-1,-1
    .endm

lista:    linea0
        .irpc i,123456#78
            linea
        .endr

longlista:    .int (-lista)/4
media:        .int 0x89ABCDEF
resto:        .int 0x01234567

formato:      .ascii "media=      %8d \t resto=  %8d \n" #formato para 4 nums
              .ascii "hexadecimal 0x%08x \t resto=0x%08x \n\0" #med/resto dec/hex

.section .text
#_start:      .global _start
main: .global main

    mov  $lista, %ebx
    mov  longlista, %ecx
    call suma
    mov  %eax, media
    mov  %edx, resto

    push resto
    push media
    push resto
    push media
    push $formato
    call printf
    add  $20, %esp #retrocedes en la pila hasta llegar a las direcciones anteriores a meterle el formato

    mov  $1, %eax
    mov  $0, %ebx
    int  $0x80
```

suma:

```
mov $0, %edi  
mov $0, %ebp  
mov $0, %esi
```

bucle:

```
mov (%ebx,%esi,4), %eax  
cld  
add %eax, %edi  
adc %edx, %ebp  
inc %esi  
cmp %esi,%ecx  
jne bucle
```

```
mov %edi, %eax  
mov %ebp, %edx
```

```
idiv %ecx  
ret
```

### Tabla de resultados

Valores	Media	Resto
1, -2...	0	-16
1,2,-3,-4...	-1	0
0x7fffffff...	0x7fffffff	0x00000000
0x80000000...	0x80000000	0x00000000
0xf0000000, 0xe0...,0xe0...,0xd0...,0xf0...	0xe0000000	0x00000000
-1...	0xffffffff	0x00000000
0, -1,-1,-1...	0	-31
0, -2,-1,-1...	-1	0
16,-1,-1,-1...	0	-16
60, -1,-1,-1...	0	28
90, -1,-1,-1...	1	26
-12, -1,-1,-1...	-1	-12