

```
.data
#Declaracion de los datos a cifrar
#Vector v dos componetes de 32 bits
v: .word 0x01234567,0x89abcdef

#Declaracion de la clave de cifrado
#Vector z 4 componetes de 32 bits
k: .word 0x11111111,0x22222222,0x33333333,0x44444444

# Declaracion constante delta

delta: .word 0x9e3779b9

res: .space 4

.text

main:

#Acceso 64 bits a cifrar

la $t2,v # t2 = direccion incio de datos a cifrar v (64 bits)
lw $t4,0($t2) # t4 = v[0] 32 bits
lw $t5,4($t2) # t5 = v[1] 32 bits

#Acceso clave de cifrado 128 bits

la $t2,k # t2 = direccion incio de clave de cifrado (128 bits)
lw $t6,0($t2) # t6 = k[0] 32 bits
lw $t7,4($t2) # t7 = k[1] 32 bits
lw $t8,8($t2) # t8 = k[1] 32 bits
lw $t9,12($t2) # t9 = k[1] 32 bits

# Acceso valor constante delta

la $t2,delta # $t2=dirección de la cosntante delta
lw $t3,0($t2) #Carga delta en t3 -> t3=delta

#Inicializacion Suma = 0
li $v0,0

#Inicializacion contador bucle
li $t0,0 # $t0=0
li $t1,32 # $t1=32

#Bucle 32 iteraciones
para: bgt $t0,$t1,finpara #si $t0>$t1 saltar finpara
#Ciclo basico

# suma= suma + delta
add $v0,$v0,$t3

# y += ((z<<4)+k[0]) xor (z+sum) xor ((z>>5)+k[1]);
# z += ((y<<4)+k[2]) xor (y+sum) xor ((y>>5)+k[3]);

#Incrementa contador bucle
addi $t0,$t0, 1 # $t0=$t0+1
j para #saltar a bucle
finpara:

# v [0]=y; v [1]=z;

fin: li $v0, 10
# Fin programa
syscall
```