

Este examen consiste en escribir un programa en el lenguaje ensamblador del MIPS y ejecutarlo usando QTspim.

Se dispone de un máximo de 55 minutos para su realización y se exigirá para aprobar que funcione correctamente para cualquier conjunto de datos.

El primer apartado es obligatorio y tiene un valor de 5 puntos.

Los otros apartados son opcionales, teniendo el segundo un valor de 3 puntos y el tercero un valor de 2 puntos.

El programa debe implementar los siguientes puntos:

- Almacenar en memoria un vector de 10 elementos compuesto por las siguientes palabras: 9, 2, 0, 9, 0, 4, 3, 6, 5, 8.
- Reservar en memoria una palabra adicional para almacenar el resultado.
- Recorrer el vector calculando la suma de los elementos que sean mayores a 2 y que además sean pares o múltiplos de 3.
- Almacenar la suma en memoria.

*suma = 0*

*valor = 2*

*para (i desde 1 hasta 10) hacer*

*si (V[i] > valor) y ((V[i] es par) o (V[i] es múltiplo de 3))*

*suma = suma + V[i]*

*finsi*

*finpara*

*guardar suma en memoria*

2. Modificar el programa para que *valor*, en lugar de estar almacenado en memoria, se introduzca por el teclado con el valor 3 y al final imprima por pantalla el siguiente mensaje:

"La suma de los elementos es: " Suma

3. Modificar el programa para que utilice medias palabras en lugar de palabras:

- El vector de 10 elementos almacenado en memoria estará compuesto por medias palabras en lugar de por palabras.
- Se reservará una media palabra para almacenar la suma.

```

.data
v: .word 9,2,0,9,0,4,3,6,5,8
suma: .space 4
.text
main: add $t0,$0,$0      # t0 (suma) = 0
      li $t1,2           # t1 (valor) = 2
      li $t2,10          # t2 (número elementos) = 10
      li $t3,1           # t3 (índice) = 1
      li $t4,1           # t4 comprueba si es par
      li $t5,3           # t5 múltiplo de 3
      la $t6,v           # t6 dirección del vector v

buc:   bgt $t3,$t2,fin    # s0 almacena los elementos del vector
      lw $s0,0($t6)      # V[i] > valor

      ble $s0,$t1,finsi  # V[i] > valor

      and $s1, $s0, $t4
      beq $0,$s1,entonces # V[i] par

      div $s0, $t5
      mfhi $s1
      bgtz $s1, fin      # V[i] múltiplo de 3

entonces: add $t0,$t0,$s0
finsi:    addi $t3,$t3,1
          addi $t6,$t6,4
          j buc
fin:      sw $t0,suma
          jr $31

```

### Apartados 2 y 3

```

.data
v: .half 9,2,0,9,0,4,3,6,5,8
suma: .space 2
frase1: .asciiz "Introduzca el valor: "
frase2: .asciiz "La suma de los elementos es: "

.text
main: add $t0,$0,$0      # t0 (suma) = 0
      li $t2,10          # t2 (número elementos) = 10
      li $t3,1           # t3 (índice) = 1
      li $t4,1           # t4 comprueba si es par
      li $t5,3           # t5 múltiplo de 3
      la $t6,v           # t6 dirección del vector v

      addi $v0,$0,4
      la $a0,frase1
      syscall
      addi $v0,$0,5
      syscall
      add $t1,$0,$v0     # t1 (valor)

```

```

buc:      bgt $t3,$t2,fin
          lh $s0,0($t6)          # s0 almacena los elementos del vector

          ble $s0,$t1,finsi      # V[i] > valor

          and $s1, $s0, $t4
          beq $0,$s1,entonces    # V[i] par

          div $s0, $t5
          mfhi $s1
          bgtz $s1, finsi        # V[i] múltiplo de 3

entonces: add $t0,$t0,$s0
finsi:    addi $t3,$t3,1
          addi $t6,$t6,2
          j buc
fin:      sw $t0,suma
          addi $v0,$0,4
          la $a0,frase2
          syscall
          add $a0,$0,$t0
          addi $v0,$0,1
          syscall
          jr $31

```