

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

Se dispone de un máximo de 1 hora 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 6 puntos. Los otros apartados son opcionales, teniendo el segundo un valor de 2 puntos y el tercero un valor de 2 puntos.

1. Escribir un programa que implemente el comportamiento descrito a continuación:

- Almacenar en memoria el vector: 3,2,4,5,6,1. Cada elemento ocupa un **byte**.
- Reservar espacio en memoria para una **palabra** en la que almacenar el **resultado**.
- Obtener el primer elemento
- El primer elemento se multiplica con aquellos elementos que sean mayores que 3 y menores a 6. Estos productos se acumulan en resultado. (**resultado=3*4+3*5=27**)
- **El programa ha de servir para cualquier vector.**

En este caso particular: **resultado=3*4+3*5=27**

```
res = 0
a=V[1]

para (i desde 2 hasta 6) hacer
    si (V[i] > 3) y (V[i] <6)
        res = res + a*V[i]
finpara

guardar res en memoria
```

2. Modificar el programa para que:

- Lea el vector por el teclado

3. Modificar el programa para que:

- Al final de su ejecución, imprima por pantalla el siguiente mensaje:
El resultado es: *res*

| Formato R | | | | | |
|----------------|---------|---------|---------|---------|--------------|
| operación (6b) | rs (5b) | rt (5b) | rd (5b) | sa (5b) | función (6b) |

| Formato I | | | |
|----------------|---------|---------|--------------------------|
| operación (6b) | rs (5b) | rt (5b) | inmediato/etiqueta (16b) |

| Formato J | |
|----------------|----------------|
| operación (6b) | etiqueta (26b) |

| Formato R (op = 000000) | | Formato I | | Formato J | |
|-------------------------|---------|-----------------|--------------------|-------------|-----------|
| instrucción | función | instrucción | operación | instrucción | operación |
| add rd,rs,rt | 100000 | addi rt,rs,inm | 001000 | j etq | 000010 |
| addu rd,rs,rt | 100001 | addiu rt,rs,inm | 001001 | jal etq | 000011 |
| and rd,rs,rt | 100100 | andi rt,rs,i | 001100 | | |
| break | 001101 | beq rs,rt,etq | 000100 | | |
| div rs,rt | 011010 | bgez rs,etq | 000001 rt=00001 | | |
| divu rs,rt | 011011 | | | | |
| jalr rd,rs | 001001 | bgtz rs,etq | 000111 rt=00000 | | |
| jr rs | 001000 | | | | |
| mfhi rd | 010000 | blez rs,etq | 000110 rt=00000 | | |
| mflo rd | 010010 | | | | |
| mthi rs | 010001 | bltz rs,etq | 000001 rt=00000 | | |
| mtlo rs | 010011 | | | | |
| mult rs,rt | 011000 | bne rs,rt,etq | 000101 | | |
| multu rs,rt | 011001 | lb rt,i(rs) | 100000 | | |
| nor rd,rs,rt | 100111 | lbu rt,i(rs) | 100100 | | |
| or rd,rs,rt | 100101 | lh rt,i(rs) | 100001 | | |
| sll rd,rt,sa | 000000 | lhu rt,i(rs) | 100101 | | |
| sllv rd,rt,rs | 000100 | lui rt,i | 001111 | | |
| slt rd,rs,rt | 101010 | lw rt,i(rs) | 100011 | | |
| sltu rd,rs,rt | 101011 | lwc1 rt,i(rs) | 110001 | | |
| sra rd,rt,sa | 000011 | ori rt,rs,i | 001101 | | |
| srav rd,rt,rs | 000111 | sb rt,i(rs) | 101000 | | |
| srl rd,rt,sa | 000010 | slti rt,rs,i | 001010 | | |
| srlv rd,rt,rs | 000110 | sltiu rt,rs,i | 001011 | | |
| sub rd,rs,rt | 100010 | sh rt,i(rs) | 101001 | | |
| subu rd,rs,rt | 100011 | sw rt,i(rs) | 101011 | | |
| syscall | 001100 | swc1 rt,i(rs) | 111001 | | |
| xor rd,rs,rt | 100110 | xori rt,rs,i | 001110 | | |

| Entrada / Salida | |
|------------------|---|
| Imprimir entero | \$v0 = 1 \$a0 = entero |
| Imprimir cadena | \$v0 = 4 \$a0 = dirección |
| Leer entero | \$v0 = 5 resultado en \$v0 |
| Leer cadena | \$v0 = 8 \$a0 = dirección \$a1 = longitud |

| Registros | |
|-------------|-------------|
| \$0 | \$zero |
| \$1 | \$at |
| \$2 – \$3 | \$v0 – \$v1 |
| \$4 – \$7 | \$a0 – \$a3 |
| \$8 – \$15 | \$t0 – \$t7 |
| \$16 – \$23 | \$s0 – \$s7 |
| \$24 – \$25 | \$t8 – \$t9 |
| \$26 – \$27 | \$k0 – \$k1 |
| \$28 | \$gp |
| \$29 | \$sp |
| \$30 | \$fp |
| \$31 | \$ra |