

# Lista 1

## Arquitetura e Organização de Computadores

Março 2024

1. Dentre as alternativas, qual programa em MIPS melhor traduz o código em C abaixo?

```
int abs(int num) {  
    if (num < 0)  
        return 0 - num;  
    else  
        return num;  
}
```

a)   abs:  
      bge \$a0, \$0, else  
      sub \$v0, \$0, \$a0  
      j abs\_exit  
      else:  
      add \$v0, \$0, \$a0  
      abs\_exit:  
      jr \$ra

b)   abs:  
      blt \$a0, \$0, else  
      sub \$v0, \$0, \$a0  
      j abs\_exit  
      else:  
      add \$v0, \$0, \$a0  
      abs\_exit:  
      jr \$ra

c)   abs:  
      bge \$a0, \$0, then  
      add \$v0, \$0, \$a0  
      j abs\_exit  
      then:  
      sub \$v0, \$0, \$a0  
      abs\_exit:  
      jr \$ra

d)   abs:  
      blt \$a0, \$s0, else  
      sub \$v0, \$s0, \$a0  
      j abs\_exit  
      else:  
      add \$v0, \$s0, \$a0  
      abs\_exit:  
      jr \$ra

2. Dentre as alternativas, quais programas em MIPS traduzem o código em C abaixo?

```
int min(int a, int b) {  
    if(a < b)  
        return a;  
    else  
        return b;  
}
```

a) min:  
blt \$a0, \$a1, then  
add \$v0, \$0, \$a0  
j min\_exit  
then:  
add \$v0, \$0, \$a1  
min\_exit:  
jr \$ra

b) min:  
add \$v0, \$0, \$a0  
blt \$a0, \$a1, min\_xt  
add \$v0, \$0, \$a1  
min\_xt:  
jr \$ra

c) min:  
blt \$a0, \$a1, then  
add \$v0, \$0, \$a1  
j min\_exit  
then:  
add \$v0, \$0, \$a0  
min\_exit:  
jr \$ra

d) min:  
bge \$a0, \$a1, then  
add \$v0, \$0, \$a1  
j min\_exit  
then:  
add \$v0, \$0, \$a0  
min\_exit:  
jr \$ra

3. Após a execução do programa abaixo, qual o estado do trecho de memória e dos registradores acessados?

```

lui $t0, 0x1001
addi $t0, $t0, 0x0008
addi $t1, $0, 0x01
sw $t1, 0($t0)
addi $t1, $0, 0x02
sw $t1, 4($t0)
addi $t1, $0, 0x0E
sw $t1, 8($t0)
addi $t1, $0, 0x10
sw $t1, 12($t0)
addi $t0, $t0, 0x0C
lw $t2, -4($t0)
lw $t3, -8($t0)
lw $t4, -12($t0)

```

a)

memória	valor	registrador	valor
0x10010000	0x00	\$a2	0x00000000
0x10010004	0x00	\$a3	0x00000000
0x10010008	0x01	\$t0	0x10010014
0x1001000C	0x02	\$t1	0x00000001
0x10010010	0x0E	\$t2	0x00000002
0x10010014	0x10	\$t3	0x0000000E
0x10010018	0x00	\$t4	0x00000010
0x1001001C	0x00	\$t5	0x00000000

b)

memória	valor	registrador	valor
0x10010000	0x00	\$a2	0x00000000
0x10010004	0x00	\$a3	0x00000000
0x10010008	0x01	\$t0	0x10010014
0x1001000C	0x02	\$t1	0x00000010
0x10010010	0x01	\$t2	0x0000000E
0x10010014	0x02	\$t3	0x00000002
0x10010018	0x0E	\$t4	0x00000001
0x1001001C	0x10	\$t5	0x00000000

c)

memória	valor	registrador	valor
0x10010000	0x00	\$a2	0x00000000
0x10010004	0x00	\$a3	0x00000000
0x10010008	0x01	\$t0	0x10010014
0x1001000C	0x02	\$t1	0x00000001
0x10010010	0x01	\$t2	0x00000002
0x10010014	0x02	\$t3	0x0000000E
0x10010018	0x0E	\$t4	0x00000010
0x1001001C	0x10	\$t5	0x00000000

d)

memória	valor	registrador	valor
0x10010000	0x00	\$a2	0x00000000
0x10010004	0x00	\$a3	0x00000000
0x10010008	0x01	\$t0	0x10010014
0x1001000C	0x02	\$t1	0x00000010
0x10010010	0x0E	\$t2	0x0000000E
0x10010014	0x10	\$t3	0x00000002
0x10010018	0x00	\$t4	0x00000001
0x1001001C	0x00	\$t5	0x00000000

4. Considere que 0xABCD, 0b1011\_1010\_1100\_0001 e 0b0100010100111110 estão armazenados em big-endian. Como é sua representação em little-endian?
- (a) 0xB3D5, 0b1000\_0011\_0101\_1101 e 0b01111110010100010
  - (b) 0xCDAB, 0b1100\_0001\_1011\_1010 e 0b00111111001000101
  - (c) 0xDCBA, 0b0001\_1100\_1010\_1011 e 0b1110001101010100
5. Para cada instrução a seguir, indique sua representação em código de máquina MIPS:
- (a) ADD \$s0, \$s1, \$s2
    - i. 0x02328020
    - ii. 0x02328022
    - iii. 0x012A4020
  - (b) J ident (considere que ident está no endereço 0x00400004)
    - i. 0x00400018
    - ii. 0x08100001
    - iii. 0x02000008
  - (c) ADDI \$t0, \$t1, 1
    - i. 0x25280001
    - ii. 0x22300001
    - iii. 0x21280001

6. Para cada código de máquina a seguir, indique a correspondente instrução em MIPS:

(a) 0x02328022

- i. and \$s1, \$s2, \$s0
- ii. add \$s2, \$s1, \$s0
- iii. sub \$s0, \$s1, \$s2

(b) 0x02000008

- i. jalr \$s0
- ii. jr \$s0
- iii. jalr \$s1

(c) 0x25280001

- i. addiu \$t0, \$t1, 1
- ii. addi \$s0, \$s1, 1
- iii. addi \$t1, \$t0, 1

Para as questões sobre representação em ponto fixo, considere a notação `fixo(c, p)`: valor armazenado em `c` bits com ponto entre o bit `p` e o bit `p+1`.

7. Qual número é armazenado como 0xA3 em `fixo(8, 3)`?

- (a) -2.7360
- (b) -5.8125
- (c) -9.5145

8. Qual valor hexadecimal representa o número -2.5 em `fixo(8, 3)`?

- (a) 0xD9
- (b) 0xEA
- (c) 0xD8

9. Qual valor hexadecimal representa o número -2.5 em `fixo(8, 0)`?

- (a) 0x8E
- (b) 0xCD
- (c) 0xFB

Exercício complementar:

10. Compile o código abaixo para MIPS. Indique os endereços e registradores correspondentes a cada variável.

```
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

int bubbleSort(int array[], int size) {

    int swaps = 0;

    for (int step = 0; step < size - 1; ++step) {
        for (int i = 0; i < size - step - 1; ++i) {
            if (array[i] > array[i + 1]) {
                swap(array + i, array + i + 1);
                swaps++;
            }
        }
    }

    return swaps;
}

int main() {
    int a[] = {2, 1, 7, 4, 3, 9, 8, 6};
    int tam = sizeof(a)/sizeof(*a);
    int s = bubbleSort(a, tam);
    return 0;
}
```