

(c) Considere a seguinte função em C.

```
void dosomething(char *s1, char *s2) {
    int i;

    for (i=0; i<strlen(s1); i++) {
        s2[i] = s1[strlen(s1)-i-1];
    }
}
```

(c.1) O que faz esta função?

(c.2) Esta função está incompleta, e por isso não implementa correctamente o comportamento desejado. Complete o código original de forma a corrigir este problema. Identifique a resposta a esta alínea através de um comentário no final de cada linha adicionada. (O comentário pode ser, por exemplo, /* resposta à alínea c.2 */))

(c.3) Altere o código original de forma que a função reserve em memória dinâmica (no *heap*) o espaço necessário para armazenar a string correspondente a s2 na função original, e devolva o endereço desta nova string. (Responda novamente no código original – serão consideradas todas as alterações que não contenham o comentário referido na alínea anterior.)

2. Assembly MIPS [8 valores]

(a) Codifique em assembly MIPS os seguintes trechos de código C. (Suponha que as variáveis **a**, **b** são do tipo int e estão armazenadas em \$s0 e \$s1, respectivamente.)

(a.1) **a = a + a + a;**

(a.2) **if (a == b) {
 a += 1;
}**
a = a+b;

Número:

Nome:

- (b) Qual o código C correspondente ao seguinte código em Assembly MIPS? Explique também em dez palavras ou menos o que faz este código.

```
bla: add $v0, $a0, $zero
      slt $t0, $a0, $zero
      bne $t0, $zero, fim
      sub $v0, $zero, $a0
fim: jr $ra
```

Explicação do código (<10 palavras):

- (c) Suponha que os autores do MIPS decidem adicionar instruções do tipo “I” à arquitectura, e desta forma são forçados a aumentar o número de bits do campo “opcode” de 6 para 7, sem alterar as características fundamentais da arquitectura como por exemplo o número e a dimensão dos registos, ou a uniformidade do campo opcode em todos os tipos de instrução. Explique quais as implicações desta alteração nas instruções de cada tipo: R, I (apenas os branches), e J. NOTA: Não chega dizer que o número de bits de certos campos das instruções passa de X para Y. Tem também de explicar as **implicações** dessa alteração no número de bits. **Quantifique** a sua resposta, por exemplo, não diga apenas que certa característica “aumenta”, mas diga também que “aumenta de <valor original> para <valor novo>”.

Tipo R:

Tipo I (considere apenas o caso dos branches):

Tipo J:

3. Mixórdia de temáticas [4 valores]

(a) Considere, após a compilação de dois ficheiros C separados, file1.c e file2.c, que o linker vai ligá-los num só executável.

a.1) As instruções do tipo “branch” têm de ser alteradas pela ligação? Porquê?

a.2) As instruções do tipo “jump” têm de ser alteradas pela ligação? Porquê?

(b) Qual o valor representado pela seguinte sequência de bits em vírgula flutuante, precisão simples: 10111111010000000000000000000000

Espaço adicional para a resolução: