

Aluno(s)	Estrutura geral, programa principal e interação com usuário (4,0 pontos)	Sub-rotinas recursivas (3,0 pontos)	Documentação: telas, comentários, explicações (3,0 pontos)	Nota
André Lisboa, Guilherme Silva e Raffael Daltoé	O programa interage com o usuário e apresenta um menu inicial. Contudo, não envia mensagem soicitando dado a inserir ou o índice do elemento a imprimir (-0,2). A pilha em alguns casos não volta para o ponto original ao final da execução. Pior, às vezes ela está mais abaixo do ponto inicial, indicando que parte da pilha de quem chamou o programa foi destruída (0x7FFFEFC0) (-0,5). Há uma linha com armazenamento desnecessário do \$ra na pilha no laço principal. O código do programa principal está misturado com o código das rotinas (-0,3). Existe uma comparação entre \$fp e \$a0 (com \$a0=0x10010004) que não faz sentido (-0,1). Os trechos de preparação de saltos para as sub-rotinas voltam com JR \$ra, mas não se entra neles com JAL (-0,2). (2,7)	Sub-rotinas em si OK. (3,0)	É desnecessário ter no relatório um referencial teórico. Contudo, uma explicação de como o programa e as sub-rotinas são estruturados seria essencial, mas não existe (-0,3). Faltou ilustrar (com telas) como a lista encadeada é criada e mantida (-0,3). (2,4)	8,10
Bruno Xavier e Drielle Teixeira	Programa principal com interação OK. Fiz um teste de inserir 3 dados ao final da lista (100, 200 e 300, nesta ordem) e depois tentei ler o primeiro, que deveria ser 100, a resposta foi 100. Depois tentei imprimir o segundo elemento da lista e o programa abortou com erro, o que indica falha de encadeamento dos elementos da lista (-1,0). O programa de fato nunca termina de forma correta, sempre com erros, pois não colocaram um syscall depois de li \$v0,10 no trecho após o rótulo encerra (-0,5). (2,5)	Rotina <code>ins_ult_el</code> parece funcionar, embora gere erros de encadeamento na lista, que depois se revelam catastróficos ao tentar fazer consultas na lista usando a rotina <code>print_nthval</code> (-1,0). (2,0)	O relatório contém apenas as respostas das 7 questões. Não explicaram a solução (-1,0). Também não colocaram telas ilustrando a operação do software (-1,0). Resposta (5) está incorreta. Como a cada chamada recursiva se empilha 12 bytes, na terceira chamada recursiva deveria haver $12 * 3 = 36$ bytes na pilha e não 16 (-0,2). (0,8)	5,30
César Munari, Felipe Lacerda e Miguel Almeida	O programa executa corretamente, até onde testei. A estrutura do heap fica correta ao final de diversas inserções e a pilha termina com mesmo valor que inicia. (4,0)	Implementação basicamente perfeita! (3,0)	Extensa e de ótima qualidade. (3,0)	10,00
Guilherme Antonini, Henrique Mrás e Thaigo Grehs	Ignoraram a especificação do trabalho e implementaram um único programa que cria uma lista encadeada e a manipula. O programa principal chama macros e não rotinas (-1,0). Mesmo praticamente sem usar a pilha, o \$sp não termina onde começa (-0,5). (2,5)	Não há sub-rotinas recursivas no programa, sequer existem sub-rotinas (pois não usaram as instruções JAL/JR \$RA) (-3,0). (0,0)	Como não respeitaram a especificação, não conseguiram responder às 7 perguntas desta (-1,0). Além disso, o relatório está bem pobre (-0,5). (1,5)	4,00
Henrique Paim, Luiz Buzzacaró e Rafael Rosa	O programa executa corretamente, até onde testei. A estrutura do heap fica correta ao final de diversas inserções e a pilha termina com mesmo valor que inicia. O código está um pouco bagunçado (-0,3), mas há uma correta separação entre programa principal e subrotinas. (3,7)	Subrotinas recursivas <code>ins_ult_el</code> e <code>print_nthval</code> estão corretas e bem implementadas. (3,0)	Documentação OK. (3,0)	9,70
Jessé Rodrigues, Mateus Rosa e Pedro Filippi	O programa executa corretamente, até onde testei. A estrutura do heap fica correta ao final de diversas inserções e a pilha termina com mesmo valor que inicia. Os autores contudo usam formas bizarras e perigosas de mover o conteúdo de um registrador para outro. Ao invés de usarem a óbvia pseudo <code>move</code> , usam a pseudo <code>la</code> de forma incorreta (mas que interessantemente funciona, quando não deveria). Em várias partes do código se faz isto usando as seguintes formas da <code>la</code> : <code>la \$t0, 0(\$v0)</code> ou <code>la \$t0,(\$t1)</code> . A forma correta seria escrever <code>move \$t7, \$v0</code> e <code>move \$t0, \$t1</code> (-0,2). (3,8)	Subrotinas recursivas <code>ins_ult_el</code> e <code>print_nthval</code> estão corretas, mas são um pouco desotimizadas, passando como parâmetros contadores desnecessários e assim alocando mais pilha que o estritamente necessário entre chamadas (-0,2). (2,8)	Documentação excelente. (3,0)	9,60
Mateus Oliveira, Matheus Schneider (Cancelou) e Nathan Espindola	Ignoraram a especificação do trabalho e implementaram um único programa que tenta criar uma lista encadeada e manipulá-la, sem muito sucesso. O programa não chama nenhuma subrotina, nem usa a pilha para nada útil (-1,0). Porquê um código fonte assim teria alguma utilidade: <code>jal loop</code> <code>loop: li \$v0, 4</code> ? Porque saltar explicitamente para alinha imediatamente seguinte? (-0,2). O programa consegue inserir elementos corretamente na lista (sem usar subrotina) mas não consegue imprimir nada de forma correta (-0,5). (2,3)	Não há sub-rotinas recursivas no programa, sequer existem sub-rotinas (usaram 3 vezes instruções JAL onde deveriam usar J [desvio simples e não para subrotina] e usaram apenas um JR \$RA, em parte do programa onde ele não faz sentido ser um retorno de sub-rotina) (-3,0). (0,0)	Como não respeitaram a especificação, não conseguiram responder às 7 perguntas desta (-1,0). Além disso, o relatório menciona subrotinas, quando não implementaram nenhuma e a documentação está muito pobre (-0,7). (1,3)	3,60