

Aluno(s)	Estrutura geral, programa principal e interação com usuário (4,0 pontos)	Rotina (3,0 pontos)	Documentação: telas, comentários, explicações (3,0 pontos)	Nota
Alan Oliveira e William Rodrigues	Código fonte desorganizado, texto que seria da rotina recursiva está inserido no meio do código do programa principal. A rotina não existe em si, pois apesar de serem usadas chamadas com JAL e de se empilhar valores (apenas na primeira chamada), não se retorna para quem chamou (instrução JR \$ra nunca é usada) e a pilha inicialmente alocada não é desalocada. A pilha é usada apenas como uma memória de rascunho em posições fixas. Na interação, após uma execução correta, às vezes solicitar novamente a opção 1 produz a mensagem "Opção invalida!!!" [sic], sendo necessário repetir a escolha, o que indica falha do algoritmo, possivelmente nas alternâncias entre dois tipos de "chamada": por exemplo, depois de rodar MDC(12,33) sempre gera o erro citado, mas depois de rodar MDC(10,25), o erro nunca ocorre. Com era de se esperar devido às falhas de implementar a subrotina, a pilha ao final da execução não volta para o seu ponto original, contendo lixo. (1,5)	Não há rotina no trabalho, do ponto de vista estrito. Apenas um conjunto complicado de laços que se comunicam via instrução JAL usada apenas como uma instrução J, sendo o valor guardado em \$RA pelo JAL absolutamente ignorado em todo o código. Há envio de valores pela pilha e manipulação destes valores "in loco" na pilha. Realmente não entenderam o conceito de subrotinas, e muito menos o de subrotinas que se comunicam via pilha. (0,8)	O que seria "instrução recursiva"? Expressão sem sentido. Mais um problema: "neste ponto é feita função recursiva", não se faz uma função, se chama ela. O relatório sintomaticamente evita abordar a questão de passagem de parâmetros e recepção de resultados, pois a implementação viola a especificação do trabalho de forma quase total. (1,0)	3,30
Arthur Silva, Drielle Teixeira (Cancelou) e Thomas Caldas	Código fonte razoavelmente organizado com rotina separada do programa principal. Há violações da especificação, contudo, pois: (1) usa-se o registrador \$t4 para passar o resultado da rotina recursiva para o programa principal; (2) a subrotina começa testando valores em registradores na sua entrada, ao invés de buscar estes valores na pilha (onde deveriam estar). Ou seja a regra de realizar a comunicação apenas usando a pilha é violada em alguns pontos. (3,6)	Rotina com algumas falhas de organização. O pior problema é usar registradores para enviar resultados para quem a chamou, ao invés de passá-los pela pilha como especificado no trabalho. Além disto fazer os testes na entrada da subrotina sobre registradores é um erro bobo, que podia ser evitado precedendo os testes das busca de atais valores na pilha. (2,5)	Pessoal, cuidem um pouco da nossa língua: "O projeto se trata do desenvolvimento"? Não, "O projeto trata do desenvolvimento", o verbo não pode ser reflexivo neste caso. Este problema aparece muitas vezes no telatório. Mais dois problemas recorrentes: "caso um dos valores for zero o programa é encerrado imediatamente para prevenir erros". Quando se usa "caso", é necessário colocar o verbo no subjuntivo e em português evitamos erros, não prevenimos (o inglês prevent se traduz por evitar). Assim, em bom português temos "caso um dos valores seja zero o programa é encerrado imediatamente para evitar erros". (2,6)	8,70
Artur Rodrigues e Douglas Carvalho	Implementação no geral boa, com pequenos problemas. O programa principal não desaloca a pilha que alocou para a chamada deixando lixo na pilha. O programa principal aloca espaço de pilha diferente das chamadas recursivas (8 contra 12 bytes), o que não é boa prática de programação. O final do programa é ditado por código que vem depois da subrotina recursiva, enquanto que o resto do programa principal está antes dela, uma má prática de programação. (3,5)	Implementação basicamente perfeita! (3,0)	De boa qualidade, mas sucinta. (2,5)	9,00
Bruno Zaffari, Murilo Daitx e Guilherme Silva	Não usaram a pilha para nada, violando a especificação do trabalho. Subrotina existe, e é chamada, mas internamente é implementada como um laço de execução violando totalmente a estrutura do código fonte C recursivo. O programa não excuta em laço, mais uma vez violando a especificação. Se um número negativo for especificado como entrada o programa entra em laço eterno. (1,8)	Não há sub-rotina recursiva no programa, violando a especificação mais global do trabalho. Não há comunicação via pilha. A subrotina é executada como um laço iterativo manipulando valores passados via registradores. (1,2)	Documentação bem pobre. (1,8)	4,80
Matheus Schneider	Não usa pilha para nada, violando a especificação do trabalho. Subrotina existe, e é chamada, mas internamente é implementada como um laço de execução violando totalmente a estrutura do código fonte C recursivo. Código fonte um pouco desorganizado, texto da rotina está inserido no meio do código do programa principal. Interação é um pouco anti-natural, pois um par de dados devem sempre ser entrados e se algum deles for -1 o programa é encerrado. (2,5)	Não há sub-rotina recursiva no programa, violando a especificação mais global do trabalho. Não há comunicação via pilha. A subrotina é executada como um laço iterativo manipulando valores passados via registradores. (1,2)	Documentação fraca. Limitou-se a apresentar as interações do programa com o usuário para fornecer valores e receber resultados. (1,5)	5,20