MAC 115 - POLI - Primeiro Semestre de 1997

AVISOS

1. Segunda prova: 6/6 (sábado). Compareçam na secretaria do Departamento de Ciência da Computação às 9 horas. A prova termina a 1 hora.
2. Cronograma para correção de EPs:
   - Quarto EP: disponível a partir de 23/6 à tarde. Devolver em 30/6.

Critério de Correção para o Terceiro EP

1 Estratégia de correção

- **(Começando a correção)** Este exercício-programa ficou difícil e longo. Para que você não tenha um trabalho excessivo, sugerimos que você comece a correção executando o programa do aluno.
- **(Falta de arquivo executável)** Pedimos para que o aluno entregasse um arquivo executável no disquete. Se ele não entregou, tente compilar o programa no Quick C (ou no compilador que você tiver a sua disposição). Caso dê erro de sintaxe ou qualquer outro problema sério na compilação, dê zero mas deixe claro na listagem que foi porque você não pode executar o programa. Entregue estes EPs separados ao seu professor.
- **(Testando cada função)** Teste uma a uma cada uma das funções abaixo, marcando as que funcionam e as que não funcionam na listagem. Você não precisa achar o erro na listagem (isso daria muito trabalho). Apenas marque que não está funcionando para determinado caso. Abaixo, dividimos o EP em pedaços e damos o valor de cada parte.

2 Funções a serem testadas e seus valores

Esperamos que vocês testem as seguintes funções.

- **(Leitura)** Caso o programa do aluno não consiga ler o arquivo de entrada que deixamos disponível no CCE (`video.txt`, `banco.txt` ou `biblio.txt`), em princípio, dê zero. Se você estiver de bom humor, veja na listagem se o aluno dá alguma explicação do que está acontecendo, e se houver alguma explicação e for possível, continue a correção seguindo alguma dica do aluno.
- **(Gravação - valor: 1.0 ponto)** Desconte até 1.0 ponto se o programa do aluno não grava corretamente.
- **(Inserção - valor: 2.0 pontos)** Desconte até 2.0 pontos se o programa não consegue inserir um novo registro, ou se a inserção tem algum problema.
- **(Leitura + Gravação + Inserção - valor: 4.0 pontos)** Estas funções estavam no `alunos.c`. Se o programa do aluno tem pelo menos estas três funções funcionando, ele deve receber pelo menos 4.0 de nota.
- **(Remoção - valor: 1.0 ponto)** Desconte até 1.0 ponto se o programa não faz remoção ou a remoção tem algum problema.

- **(Busca - valor: 1.5 pontos)** Desconte 1.5 pontos se o programa não faz busca. Se o programa acha palavras apenas no fim, pode ser porque o aluno procura inclusive o `´\n´`. Neste caso, desconte 0.5 ponto. Se o programa acha apenas o nome inteiro, por exemplo, porque usa o `strcmp`, desconte 1.0 ponto. Se o programa busca apenas em um dos vetores (titulo ou titular, por exemplo), desconte 0.5 ponto.
- **(Lista + Listatitulo - valor: 1.5 pontos)** Se o aluno implementa o `listatitulo`, mas não o `lista`, desconte 1.0 ponto. Se o `listatitulo` do aluno não pára a cada 10 registros, ou seja, primeiro ele mostra todos os registros, depois pergunta qual o usuário quer listar, não desconte nada, desde que tudo esteja funcionando. Se o `lista` mostra o registro errado, desconte 0.5 ponto.
- **(Edição + Retirada + Devolução - valor: 1.0 ponto)** Se o aluno, após a listagem de um registro, nem imprime o menu de manipulação com estas opções, desconte 1.0 ponto. Se apenas uma ou duas destas três funções está implementada, ou está correta, desconte 0.5 ponto. Se todas as três estão erradas, desconte 1.0 ponto. Note que, dependendo de como o aluno fez o `listatitulo`, a opção "continuar listagem" não faz sentido. Não desconte nada quanto a isso.
- **(Programa principal e interface com o usuário - valor: 1.0 ponto)** Desconte no máximo 1.0 ponto se a interface com o usuário é muito ruim ou o controle do programa está com problemas, ou seja, se o programa não segue o curso normal, apresentando os menus certos, voltando as telas corretamente, etc.

Escreva na listagem do aluno em quais das funções listadas acima você detectou problema, e quanto você descontou em cada caso.

3 Olhando a listagem

Dê uma olhada geral na listagem do aluno para ver se tem algo que lhe chama a atenção, e que você queira comentar.

- **(Em caso de cola)** Dê zero para EPs colados. Marque os grupos de cola e os deixe separados para que o professor tome conhecimento, como das outras vezes.
- **(Funções do enunciado)** Não desconte caso o aluno tenha mudado um pouco o cabeçalho das funções descritas no enunciado do EP. Por exemplo, o aluno pode ter mudado o tipo da função, a forma como o `n` é passado como parâmetro, etc.
- **(Indicar erros encontrados)** Os erros encontrados devem ser indicados na listagem do programa, junto com o valor descontado. Como já dissemos acima, escreva quais funções você detectou que não estavam funcionando. Anote também todas as observações que quiser fazer. A nota final do EP deve estar num lugar visível.
- **(Identação, comentários, estrutura do programa, uso de variáveis globais, etc)** Desconte no máximo 1.0 ponto por tudo isso. Este exercício estava bastante complicado para eles, por isso gostaríamos que a nota refletisse quanto trabalho o aluno teve. Por exemplo, gostaríamos que um aluno cujo programa está funcionando direitinho, ficasse com nota pelo menos 9.0, independente da aparência do programa e da listagem. Um aluno que tem quase todas as funções funcionando, deveria ficar com uma nota entre 7.0 e 9.0. Um aluno que fez mais ou menos metade das funções, com uma nota entre 5.0 e 7.0. Quanto mais perto do básico que foi dado no `alunos.c`, mais perto de 4.0 a sua nota deve estar. Se nem o que é dado no `alunos.c` o aluno fez, sua nota deve ser abaixo de 4.0.

4 Observações Finais

- **(Notas numa planilha)** Como no EP anterior, as notas devem ser passadas ou em uma planilha (de preferência a mesma) ou anotadas em uma tabela, a critério da preferência do professor.
- **(Casos difíceis)** Nos casos difíceis de avaliar, o monitor deve consultar o professor da turma.
- **(Erro de sintaxe)** Programa com erro de sintaxe deve receber zero.