MAC 115 - POLI - Primeiro Semestre de 1997

Critério de Correção para o Primeiro EP

1 Observações Gerais

- os erros encontrados devem ser indicados na listagem do programa, junto com o valor descontado. Anote também todas as observações que quiser fazer. A nota final do EP deve estar num lugar visível;
- nos casos difíceis de avaliar, o monitor deve consultar o professor da turma;
- as notas dos EP's devem ser passadas ou numa planilha ou anotadas numa tabela, a critério da preferência do professor;
- Será muito difícil dizer se houve cópia entre EP's, mas se houver indícios fortes de que algum programa foi copiado, então zero no exercício-programa! A decisão final porém é do professor. Nos EP's também deve ser anotada a nota que o aluno tiraria no EP caso não houvesse cola. Desconfie de EP's com diferenças gritantes de estilo de um exercício para outro dentro do mesmo EP.
  Como este primeiro exercício é curto e simples, é natural que apareçam soluções semelhantes. Se estiver desconfiado da "cola", corrija-os como se não houvesse esse problema e os entregue separado dos outros EP's para o professor. Você deve classificar os EP's iguais em "classes de equivalência" A, B, C, etc onde todos os EP's da classe A são semelhantes entre si, e assim por diante. Anote num lugar visível a qual classe de equivalência pertence cada EP com problema de "cola".
  Separe os programas "suspeitos" e entregue ao professor;
- Neste primeiro EP ainda não estaremos descontando nota em função do estilo (indentação, comentários). Caso julgue conveniente, escreva no programa do aluno quais aspectos de estilo ele deve aprimorar para o segundo EP. Da mesma forma, não estaremos descontando nota por falta de elegância ou concisão da solução (estratégia algorítmica, ausência de redundâncias, número reduzido de comandos, etc.);
- Não estaremos descontando valor por "consistência de dados de entrada", ou seja, o aluno não precisar analisar a validade dos dados de entrada;
- Programa com erro de sintaxe deve receber zero;
- Caso detecte uso de variável não inicializada[1], desconte 1.0 ponto;
- **Para versões "Gwonck", "Caixa Registradora" e "Números Romanos"** :
  A tarefa de correção deste primeiro EP está sendo dificultada pelo fato de que os alunos não foram instruídos sobre como jogar a saída do programa na impressora (para o segundo EP a informação já está disponível). Ou seja, na maior parte dos casos você estará recebendo apenas o disquete e a listagem do código fonte. Use seu bom senso para determinar se você deve ou não executar o programa de um aluno. Como este primeiro programa é muito simples, para programas muito bem feitos (concisos, bem indentados, com comentários) pode ser possível avaliar olhando apenas o fonte. Mas se o programa estiver difícil de ler, a melhor alternativa é carregar o programa e executá-lo.

2 Versão "GWONCK"

Algumas observações sobre esta versão do EP:

- é normal que o aluno tenha um programa meio "repetitivo", ou seja, um trecho do programa resolve o problema para a base 5, aí a mesma coisa para a base 7 e 9. (Os alunos só viram malha encaixada recentemente);

[1] Uma variável cujo valor esteja sendo usado sem que antes tenha sido lido ou atribuído

- para programas que não estejam funcionando bem, tente avaliar a quantidade de casos em que o programa não funciona (por exemplo, funciona apenas para números diferentes de zero, ou maiores que 10, etc). Para casos mais particulares o desconto pode ser menor, casos mais amplos em geral indicam um erro de lógica mais grave, que o aluno deveria ter sido capaz de detectar na fase de teste. (Dada a simplicidade deste EP, não é muito provável que os programas não funcionem para o caso geral);
- os alunos são instruídos a não usarem coisas equivalentes a "goto". (Se arrumarem um jeitode usar neste EP — não seria natural — desconte 2.0)

3 Versão "Caixa Registradora"

- é normal que o aluno tenha um programa meio "repetitivo", ou seja, que o programa seja uma repetição da mesma estrutura para notas de 500, 100, 50, etc;
- as soluções em geral apresentam muitas variáveis (algo como 25 ou mais);
- O enunciado especifica que:
  - a execução deve continuar com uma mensagem quando não for disponível dar o troco (ou seja, as notas disponíveis não são suficientes para dar o valor exato necessário);
  - a execução termina quando o usuário fornecer um pagamento (número de notas de 500, 100, etc) que é menor que o "valor a ser pago" (quantidade fornecida pelo usuário).

  Erros comuns são:

  1. Interromper a execução mesmo no caso de troco insuficiente;
  2. Quando não é possível dar o troco, o caixa fica em estado inconsistente (ou seja, foram contabilizadas as notas de entrada, o troco não foi feito e notas não foram devolvidas);
  3. O aluno lê os valores para "valor a ser pago" e "pagamento" fora do loop, ou seja, o programa dele está bem errado mesmo!
  4. O aluno calcula a quantidade de notas da seguinte forma: para 1598, conclui que são necessárias 3 notas de 500, 1 de 50, 4 de 10 e 8 de 1. Se qualquer destas quantidades não estiver disponível, o aluno já avisa que não há troco (ou seja, ignora por exemplo o fato de não haver nota de 500 pode estar ok, dependendo de quantas notas de valores inferiores existem).
- os alunos são instruídos a não usarem coisas equivalentes a "goto". Desconte 2.0 caso encontre algo assim.

4 Versão "Números Romanos"

- É normal que os alunos façam programas longos e repetitivos, já que aprenderam malhas encaixadas apenas recentemente;
- Os alunos foram instruídos a não lerem dígito a dígito. Caso façam isso, -4.0;
- Os alunos foram instruídos a n ao usarem vetores. Caso façam isso, -1.0;
- Caso o programa do aluno só funcione para números no intervalo indicado no enunciado do EP, isso deve ser indicado na listagem, ainda que sem descontar pontos;
- É possível que apareçam soluções sem nenhum laço:

```
scanf("%d",&numero);
if( numero >= 4000 ) printf("MMMM");
else if(numero >= 3000 ) printf("MMM");
else if(numero >= 2000 ) printf("MM");
else if ...
```

Neste caso indique que a solução não é boa, mas não desconte pontos.

- os alunos são instruídos a não usarem coisas equivalentes a “goto”. (Se arrumarem um jeitode usar neste EP — não seria natural — desconte 2.0)

5 Versão “Caixa Eletrônico”

- Esta versão apresenta informação sobre como fazer saída na impressora. Caso o aluno não entregue junto a cópia impressa da saída de seu programa, desconte 1.0.
- as soluções em geral apresentam muitas variáveis (algo como 25 ou mais);
- é normal que o aluno tenha um programa meio “repetitivo”, ou seja, que o programa seja uma repetição da mesma estrutura para notas de 100, 50, 10, 5 e 1;
- As soluções têm, em geral, três partes:
    1. caso a transação seja depósito
       Se esta parte estiver errada, desconte 1.5
    2. caso a transação seja saque e não haja dinheiro
       Se esta parte estiver errada, desconte 1.0
    3. caso a transação seja saque e não haja notas suficientes
       Se esta parte estiver errada, desconte 1.0
    4. caso a transação seja saque e há notas disponíveis
       Se esta parte estiver errada, desconte 1.5
- Quando não é possível dar o troco, o caixa deve ficar em estado consistente (ou seja, o que estava antes da transação). Um erro possível é que o aluno vá descontando notas e em algum ponto descubra que não há notas suficientes, e acabe não repondo as notas já descontadas.
- O aluno lê os valores da transação fora do loop, ou seja, o programa dele está bem errado mesmo!
- O aluno calcula a quantidade de notas da seguinte forma: para 1598, conclui que são necessárias 15 notas de 100, 1 de 50, 4 de 10, 1 de 5 e 3 de 1. Se qualquer destas quantidades não estiver disponível, o aluno já avisa que não há troco (ou seja, ignora por exemplo o fato de não haver nota de 100 pode estar ok, dependendo de quantas notas de valores inferiores existem).