

Programação Imperativa

LEI (1º ano)– Ano Lectivo 2006/07

3ª fase de avaliação prática

Problemas A + B + C + D

6 de Junho de 2007

Como já sabe, Hipólito Zacarias e a sua mulher são um casal muito amigo mas ambos muito teimosos; porém, felizmente, gostam muito de fazer apostas e assim entretêm os seus dias.... só que precisam da sua ajuda para resolver os conflitos.

Para testar a sua interacção com o sistema Mooshak, sugere-se que comece por resolver o problema de teste A.

A. Já sou perito nisto de concorrer no Mooshak

Escreva um programa, e codifique-o em C, para ler do `stdin` a idade e a altura do Hipólito e a idade e altura da sua senhora e enviar para o `stdout` qual é o mais alto ("ele" ou "ela" ou "empate"), e qual a idade média do casal (número real com 1 casa decimal), escrevendo um resultado por linha.

Exemplo:

```
stdin          stdout
72 1.65 78 1.60  ele
                75.0
```

B. A Arca de Jogos

O casal Zacarias está fascinado com a Arca de Jogos desenvolvida pela turma LEI1&C.ia. Compraram 2 cópias e passam o dia a jogar (cada um no seu Mac), anotando numa folha, por ordem crescente, os pontos que vão obtendo em cada jogada. Agora querem saber quem ganhou e perceber as pontuações relativas.

Para os ajudar a fazer a classificação final escreva um programa, e codifique-o em C, para ler do `stdin` a sequência de pontos da Sra. de Zacarias seguida da sequência de pontos do Hipólito (ambas terminadas por -1, garantindo-se que não há pontuações repetidas), e enviar para o `stdout` uma lista única com os pontos por ordem decrescente. Cada linha da saída conterá uma letra maiúscula ("M" de maria, ou "H" de Hipólito) seguida da pontuação, separadas por 1 espaço.

Exemplo:

```
stdin          stdout
20 25 -1       H 40
12 40 -1       M 25
                M 20
                H 12
```

C. Brincando com Palavras

Para ajudar a entreter o casal Zacarias, quer-se criar um novo jogo de palavras. Para isso escreva um programa, e codifique-o em C, para ler do `stdin` uma lista de palavras (cada palavra terá no máximo 50 caracteres) e reproduzi-la no `stdout` (uma palavra por linha) de tal forma que cada palavra de saída se obtém de uma palavra de entrada retirando todas as vogais, excepto a vogal que comece a palavra. As palavras devem aparecer agrupadas por ordem crescente do seu comprimento original. No caso de duas palavras com o mesmo tamanho, devem ser ordenadas alfabeticamente

Exemplo:

<code>stdin</code>	<code>stdout</code>
ananas	els
banana	anns
chaleira	bnn
dormitorio	chlr
elisa	drmtr

D. Calculando em Polaco

Hipólito está em pânico: descobriu que se estiver a ler uma expressão aritmética da esquerda para a direita e for calculando as operações (adição, subtração, multiplicação, divisão) logo que aparecem os dois operandos (números reais), não obtém o resultado correcto porque não observa a prioridade das operações.

Então Maria explica-lhe que se a expressão aparecer em *notação polaca*, ou *post-fixa*, segundo a qual um operador aparece sempre depois dos dois operandos a que se aplica, já pode fazer o cálculo à medida que lê a expressão. Maria acrescenta, basta que vás empilhando os operandos (OPER) até que apareça um operador (OP); nessa altura desempilhas os 2 operandos do topo, fazes a operação em causa e empilhas o resultado; quando chegares ao fim da leitura, o único valor no topo é o resultado final.

Para ajudar a Sra. de Zacarias a ilustrar a ideia a seu *Ás da Matemática*, escreva um programa, e codifique-o em C, para ler do `stdin` uma lista de operandos (cada qual virá precedido pela palavra OPER) e operadores (cada qual precedido pela palavra OP) e, após ler o sinal IGUAL, escrever no `stdout` o resultado da expressão, calculado segundo as regras acima. Os operadores serão denotados pelas palavras: ADI, SUB, MUL e DIV.

Exemplo (pretende-se calcular $12/(6 - 3) + 12/6 - 3$):

<code>stdin</code>	<code>stdout</code>
OPER 12	3.0
OPER 6	
OPER 3	
OP SUB	
OP DIV	
OPER 12	
OPER 6	
OP DIV	
OP ADI	
OPER 3	
OP SUB	
IGUAL	