

Warmup – Sarah Menezes

3ª Seletiva Interna – 2012/2

Sevidor BOCA:

<http://10.20.107.205/boca/>
(acesso interno)

<http://200.19.107.205/boca/>
(acesso externo)



Organizadores:

Claudio Cesar de Sá (coordenação geral), Lucas Negri (coordenação técnica), Yuri Kaszubowski Lopes (coordenação técnica), Alexandre Gonçalves Silva, Roberto Silvio Ubertino Rosso Jr., André Luiz Guedes (Dinf/UFPr – revisão técnica), Fernando Deeke Sasse

Lembretes:

- Aos *javaneiros*: o nome da classe deve ser o mesmo nome do arquivo a ser submetido. Ex: classe `petrus`, nome do arquivo `petrus.java`;
- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova;
- A correção é automatizada, portanto, siga atentamente as exigências da tarefa quanto ao formato da entrada e saída de seu programa. Deve-se considerar entradas e saídas padrão;
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. As soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas;
- Teste seu programa antes de submetê-lo. A cada problema detectado (erro de compilação, erro em tempo de execução, solução incorreta, formatação imprecisa, tempo excedido ...), há penalização de 20 minutos. O tempo é critério de desempate entre duas ou mais equipes com a mesma quantidade de problemas resolvidos;
- Utilize o *clarification* para dúvidas da prova. Os juízes podem opcionalmente atendê-lo com respostas acessíveis a todos;
- A interface KDE está disponível nas máquinas Linux, que pode ser utilizada ao invés da Unity. Para isto, basta dar *logout*, e selecionar a interface KDE. Usuário e senha: *udesc*;
- O nome *Sarah Menezes* é uma homenagem a primeira judoca brasileira a ganhar uma medalha de ouro nas olimpíadas de Londres.

1 Problema A: Agora eu Passo!

Arquivo: `agoravai.[c|cpp|java]`

O prof. Claudiusvirus sempre está procurando inventar uma moda. Talvez seja por isto que ele decidiu ser pesquisador–professor, mas isto é uma outra história.

Procurando ser realista com suas disciplinas, ele decidiu instituir o *provão final* de disciplina para todos alunos do curso. A idéia eliminar o exame final, ao qual a instituição o obriga em aplicar, e os alunos odeiam realizar.

Assim, a média da aprovação da disciplina passou a ser 5,0 ao invés de 7,0. Assim, ao final do semestre o professor aplica o seu *provão final*, o qual tem um peso significativo sobre a média semestral (MS).

O professor quer a sua ajuda, para que você implemente a sua fórmula de aprovação, vista na Equação 1.

$$MS = p_1n_1 + p_2n_2 + \cdots + p_nn_n + p_{provao} \cdot n_{provao} \quad (1)$$

Cada p_i é peso da nota n_i , definido por algum valor no intervalo $0.0 \leq p_i \leq 1.0$, sendo que a soma dos pesos das notas comuns é sempre menor do que 1, ou seja, $p_{provao} > 0$. A nota n_i é definida por algum valor no intervalo $0 \leq n_i \leq 10.0$. O peso do provão é dado por pela Equação 2.

$$p_{provao} = 1 - \sum_{i=1}^n p_i \quad (2)$$

A saída do programa é constituída por uma linha, contendo **aprovado** se $MS \geq 5.0$ ou **volte no exame** caso contrário.

Especificação da Entrada

A primeira linha de entrada consiste em um único inteiro N ($1 \leq N \leq 10$), a quantidade de avaliações incluindo o provão. A linha seguinte contém $N - 1$ pares de nota e peso, seguida pela nota no provão. O peso do provão deve ser obtido pela 2. Um valor 0 indica que acabaram as entradas.

Especificação da Saída

Para cada caso de teste, imprima a seguinte mensagem em uma linha:

1. **aprovado** se $MS \geq 5.0$
2. **volte no exame** caso contrário

Exemplo de entrada

```
3
2 0.4 5 0.3 8
2
2 0.1 9
0
```

Exemplo de saída

```
volte no exame
aprovado
```

2 Problema B: Controle de Natalidade

Arquivo: uga.[c|cpp|java]

Na república de Uga-Uga institui-se o controle de natalidade. A regra para o contador Og, barnabé do ministério da saúde, era simples: no máximo, cada casal poderia ter até 10 filhos, distribuídos no máximo em 5 meninas e 5 meninos.

Sim, uma regra simples em face de cada mão ter 5 dedos. A mão esquerda é usada para contar as meninas, e a mão direita os meninos. Og estava feliz com isto.

Sua missão: escrever um programa para ajudar o Og a ler os dados de cada mão e imprimir o total de filhos de cada casa.

Especificação da Entrada

A entrada contém vários casos de teste. Cada caso de teste consiste de uma única linha contendo dois inteiros **E** e **D**, separadas por um espaço simples, indicando respectivamente o número de filhas e filhos. Para o caso valores válidos a serem somados: $0 \leq E \leq 5$ e $0 \leq D \leq 5$.

Caso algum destes valores sejam negativos, algo inviável para o homem das cavernas, pois os negativos foram descobertos após esta época, trata-se aí de um problema NP, leia-se, um problema com solubilidade: “**Nem a Pedra**”.

O final da entrada é indicado por $E = D = 0$. A entrada deve ser lida da entrada padrão.

Especificação da Saída

Para cada caso de teste na entrada imprima uma linha contendo um único inteiro indicando quantos filhos tem caverna. Caso algum destes valores sejam negativos, seu programa deve imprimir a indignação de Og: Og diz: Eh um NP! (o símbolo ‘!’ ao final da frase é de indignação.).

A saída deve ser escrita na saída padrão.

Exemplo de Entrada

```
3 4
5 -5
5 5
-1 -1
0 5
5 0
0 0
```

Exemplo de Saída

```
7
Og diz: Eh um NP!
10
Og diz: Eh um NP!
5
5
```