

Warmup

1^a Maratona Catarinense de Programação

Sevidor BOCA:

<http://10.20.107.205/>
(Seletiva UDESC)

<http://10.19.107.207/>
(1^a MCP)



Organizadores:

Claudio Cesar de Sá (coordenação geral), Lucas Hermann Negri (coordenação técnica), Yuri Kaszubowski Lopes, Marlon Fernandes de Alcântara, Alexandre Gonçalves Silva, Roberto Silvio Ubertino Rosso Jr., Rogério Eduardo da Silva

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 no lugar da Unity. Para isto, basta dar *logout*, e selecionar a interface KDE. Usuário e senha: *udesc*;

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 ideia é 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.

$$MS = p_1 n_1 + p_2 n_2 + \dots + p_n n_n + p_{\text{provao}} \cdot n_{\text{provao}} \quad (1)$$

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, $p_{\text{provao}} > 0$. 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.

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

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: Será que ele avança?

Arquivo: salto.[c|cpp|java]

Ranoberto e Ranubia são duas alegres rãs adolescentes que moram às margens do rio Cubatão, na Serra Dona Francisca. Ranoberto observou as longas pernas saltadoras de Ranubia e decidiu que quer conhecê-la melhor. Como ele é um pouco tímido e não sabe bem como iniciar a conversa, pensou em convidá-la para um jogo divertido, o que pode facilitar esta paquera. Ranubia gostou do estilo dele e aceitou o convite.

O jogo chama-se “Rã Saltadora”: a partir de posições iniciais eles vão saltar um sobre o outro alternadamente. Ambos, Ranoberto e Ranubia, são capazes de saltar há uma distância horizontal máxima de até **10** unidades, em cada salto simples.

Você recebeu uma lista de posições válidas onde Ranoberto e Ranubia podem se posicionar: $x_1 x_2 \dots x_n$. Como Ranoberto é um cavalheiro deixará para Ranubia o primeiro salto. Ranubia começa inicialmente na posição x_1 e Ranoberto começa inicialmente na posição x_2 ; o objetivo deles é alcançar a posição x_n . Determine o número mínimo de saltos necessários para que cada um, Ranoberto ou Ranubia, alcance o objetivo. Aos dois jogadores não é permitido permanecer na mesma posição ao mesmo tempo (afinal, eles ainda estão se conhecendo...), e em cada salto, o jogador que estiver atrás deve pular por cima do jogador à frente.

Entrada

O arquivo de entrada contém múltiplos casos de teste. Cada caso de teste iniciará com uma linha simples contendo um inteiro simples n (onde $2 \leq n \leq 100000$). A linha seguinte irá conter uma lista de inteiros $x_1 x_2 \dots x_n$, onde $0 \leq x_1 < x_2 < \dots < x_n \leq 1000000$. O final de arquivo é denotado por uma linha simples contendo 0.

Saída

Para cada caso de teste de entrada, imprima o número total de saltos mínimo necessários para que um dos jogadores, ou Ranoberto ou Ranubia, chegue ao destino. Caso nenhum possa alcançar o destino, imprima -1.

Exemplo de Entrada

```
6
3 5 9 12 15 17
6
3 5 9 12 30 40
2
3 5
7
0 1 7 8 11 15 19
0
```

Exemplo de Saída

```
3
-1
0
4
```