

# I Maratona SBC de Programação do Cerrado: Aquecimento

## Caderno de Problemas

05 de abril de 2025



**1ª Maratona SBC do Cerrado**

(Este caderno contém 3 problemas)

### **Comissão Organizadora:**

Alberto Tavares Duarte Neto (UnB)  
Bruno César Ribas (UnB/FCTE)  
Daniel Porto (UnB)  
Daniel Saad Nogueira Nunes (IFB)  
Edson Alves das Costa Júnior (UnB/FCTE)  
Gabriel Bernardi (UFU/Neospace)  
Giullia Rodrigues de Menezes (UFU)  
Guilherme Novaes Ramos (UnB)  
Gustavo Machado Leal (UFG)  
Jeremias Moreira Gomes (IDP)  
João Henrique de Souza Pereira (UFU/Neospace)  
José Leite (UnBallon)  
Luiz Cláudio Theodoro (UFU)  
Maxwell Oliveira (UnB)  
Tiago de Souza Fernandes (UnBallon)

Universidade Federal de Uberlândia, Campus Sta. Mônica, Uberlândia, Av. João Naves de  
Ávila, 2121, MG

## Lembretes

- É permitido consultar livros, anotações ou qualquer outro material impresso durante a prova, entretanto, o mesmo não vale para materiais dispostos eletronicamente.
- A correção é automatizada, portanto, siga atentamente as exigências da tarefa quanto ao formato da entrada e saída conforme as amostras dos exemplos. Deve-se considerar entradas e saídas padrões;
- Para cada problema, além dos testes públicos, o juiz executará a sua submissão contra uma série de testes secretos para fornecer um parecer sobre a correção do programa.
- Procure resolver o problema de maneira eficiente. Se o tempo superar o limite pré-definido, a solução não é aceita. Lembre-se que as soluções são testadas com outras entradas além das apresentadas como exemplo dos problemas;
- Utilize a aba *clarification* para dúvidas da prova. Os juízes podem opcionalmente atendê-lo com respostas acessíveis a todos;

## C/C++

- Seu programa deve retornar zero, executando, como último comando, `return 0` ou `exit 0`.

## Java

- Não declare ‘`package`’ no seu programa Java.
- Note que a convenção para o nome do arquivo-fonte deve ser obedecida, o que significa que o nome de sua classe pública deve ser uma letra maiúscula igual à letra que identifica o problema.

## Problema A – IMC aproximado

**Limite de tempo: 1s**  
**Limite de memória: 256MB**

Autor: Edson Alves

O índice de massa corpórea (IMC) é um dos indicadores de saúde de um indivíduo, computado a partir de sua massa  $m$ , em quilos, e sua altura  $h$ , em metros, por meio da relação

$$IMC = \frac{m}{h^2}$$

O intervalo  $[18.5, 24.9]$  é considerado a faixa de normalidade: valores de IMC abaixo ou acima desta faixa correspondem a magreza ou a obesidade, respectivamente.

Uma recomendação informal é que, para estar dentro da faixa de normalidade, basta ao indivíduo ter massa igual ao descarte da parte inteira de sua altura. Por exemplo, se o indivíduo tem  $1.70 m$  de altura, ele deveria ter massa igual a  $70 kg$ .

Dada a altura do indivíduo, em metros, determine se esta aproximação leva ou não a um IMC dentro da faixa da normalidade.

### Entrada

A entrada consiste em uma única linha contendo a altura  $h$  ( $1.01 \leq h \leq 1.99$ ) do indivíduo, em metros. É garantido que a altura será dada com exatamente duas casas decimais.

### Saída

Se o IMC resultante da aproximação descrita estiver dentro da faixa de normalidade, imprima, em uma linha, a mensagem “Sim”. Caso contrário, imprima “Nao” (sem acento e sem aspas).

### Exemplo

Entrada	Saída
1.70	Sim
1.92	Nao
1.16	Nao

### Notas

No primeiro caso, o IMC aproximado seria igual  $24.22$ , dentro da faixa da normalidade.

No segundo caso, o valor aproximado do IMC está acima da faixa de normalidade ( $24.95$ ).

No terceiro caso, a aproximação levaria à magreza ( $11.89$ ).

## Problema B – Ticket de Estacionamento

**Limite de tempo: 2s**  
**Limite de memória: 256MB**

Autor: Daniel Saad Nogueira Nunes

Um estacionamento possui o seguinte sistema de tarifas:

Tempo	Tarifa
Primeiros quinze minutos	Grátis
Primeira hora	R\$ 0,10 por minuto.
Até a terceira hora	R\$ 0,08 por minuto.
Até a sétima hora	R\$ 0,06 por minuto.
Demais horas	R\$ 0,02 por minuto.

Este sistema é regressivo, isto é, supondo que um usuário passe 2 horas no estacionamento, dessas duas horas: 15 minutos são gratuitos, 10 centavos são aplicados para cada minuto dos 45 minutos da primeira hora, e 8 centavos são aplicados para cada minuto da hora restante.

Apesar de tudo, este estacionamento está desprovido do sistema que efetua o cálculo de quanto o usuário do estacionamento deve pagar, e por isso, o proprietário do estacionamento contratou você para criar um sistema que, dada a quantidade de minutos que um usuário utilizou o estacionamento, calcular a quantia a ser paga por ele.

### Entrada

A primeira linha da entrada possui um inteiro  $N$  ( $1 \leq N \leq 10^5$ ) indicando a quantidade de usuários.

As próximas  $N$  linhas descrevem os tempos utilizados por cada usuário, isto é, a  $i$ -ésima linha deste conjunto contém um inteiro  $T_i$  ( $1 \leq T_i \leq 10^5$ ), indicando a quantidade de minutos que o  $i$ -ésimo usuário utilizou.

### Saída

Para cada caso de teste, seu programa deverá imprimir uma linha com o valor a ser pago pelo usuário com duas casa decimais de precisão.

### Exemplo

Entrada	Saída
6	0.00
10	1.50
30	7.70
100	15.30
200	27.30
400	36.10
800	

---

## Problema C – Run-twice

**Limite de tempo: 1s**  
**Limite de memória: 256MB**

Autor: Alberto Neto

Esse é um problema de **dupla execução**.

Você, agente secreto da SBC (Sociedade de Busca e Criptografia), recebe uma missão do agente Carlinhos para mandar arquivos secretos para João Henrique. Esta informação é representada como um inteiro  $x$ . O seu trabalho é escrever funções que criptografem e descriptografem esta informação.

O seu código será executado duas vezes. Primeiro, você recebe um inteiro  $m$ , que tem valor 1 se é para criptografar ou 2 se é para descriptografar.

Se  $m = 1$ , então você recebe a informação  $x$ , e deve imprimir um código  $c$ . Para que sua criptografia não seja muito fraca, é necessário que  $x$  seja **diferente** de  $c$ .

Se  $m = 2$ , você recebe o código  $c$  gerado na execução passada e deve recuperar a informação  $x$ .

Escreva o programa pedido e salve o Cerrado.

### Entrada

A primeira linha de entrada contém um inteiro  $m$  ( $1 \leq m \leq 2$ ) —  $m$  é 1 se é a primeira execução, ou 2 se é a segunda execução.

Na primeira execução, a segunda linha de entrada contém um único inteiro  $x$  ( $0 \leq x \leq 1000$ ) — o número a ser codificado.

Na segunda execução, a segunda linha de entrada contém um único inteiro  $c$  ( $0 \leq c \leq 2000$ ;  $c \neq x$ ) — o código gerado pela sua primeira execução.

### Saída

Na primeira execução, imprima um inteiro  $c$  **diferente de  $x$**  ( $0 \leq c \leq 2000$ ;  $c \neq x$ ). Este código fará parte da entrada da segunda execução.

Na segunda execução, imprima um inteiro  $r$ . A resposta será considerada correta se  $r = x$ .

### Interação

Explicação de um problema **dupla execução**.

O seu código será executado duas vezes. Na primeira, a entrada será um caso de teste do juiz. A saída do seu código na primeira execução será validada e, se estiver no formato correto, uma nova execução do seu código será feita cuja entrada é a saída da primeira execução.

A resposta será considerada correta se a saída da segunda execução satisfizer as exigências do problema.

### Exemplo

Entrada	Saída
1	2
2	
1	314
314	