



Professor: Samuel Martins (samuel.martins@ifsp.edu.br)

Prêmio de 1 Milhão

1. Descrição

Alice e Bia criaram uma página na Internet com informações sobre o Macaco-prego-de-peito-amarelo, uma espécie em extinção. A página mostra como todos podem ajudar a manter o habitat natural para evitar que a espécie seja extinta.



Uma empresa gostou tanto da iniciativa de Alice e Bia que prometeu doar um prêmio para que as duas amigas possam realizar outras iniciativas semelhantes. A empresa decidiu que o prêmio seria dado quando a **soma do número de acessos à página chegasse a 1 milhão**.

Dada a lista de acessos diários que ocorreram à página de Alice e Bia, escreva um programa para determinar quantos dias foram necessários para a soma dos acessos chegar a 1 milhão e as amigas ganharem o prêmio. A soma de todos os valores da lista é maior do que ou igual a 1 milhão (ou seja, Alice e Bia certamente ganham o prêmio).

2. Especificação da Entrada e Saída

Entrada

A primeira linha da entrada contém um **número inteiro** N ($0 \leq N \leq 1000$), que indica o número de dias que a lista contém.

Cada uma das linhas seguintes contém **um único inteiro** A ($0 \leq A \leq 1000000$), que consiste do número de acessos em um dia. O primeiro número dado indica o número de acessos no primeiro dia, o segundo número dado indica o número de acessos no segundo dia, e assim por diante.

Saída

Seu programa deve escrever na saída **uma única linha**, contendo um único número inteiro que corresponde ao número de dias que foram necessários para a soma dos acessos à página de Alice e Bia chegar a 1 milhão.

Exemplos

Entrada	Saída
5	4
100	
99900	
400000	
500000	
600000	

Entrada	Saída
1	1
1000000	

Entrada	Saída
10	10
39349	
28815	
41815	
10659	
118619	
148236	
87930	
118351	
33654	
372572	

3. Observações Gerais

- **Apenas um integrante** da dupla deverá submeter o código;
- Caso ambos submetam, será o considerado o código da última submissão;
- A nota é dada pelo **número de casos de teste acertados**;
- Codigos com **erros de compilação e execução**, tais como Segmentation Fault, **serão considerados errados**;
- Utilize **return 0;** na main de seu programa;
- Qualquer tentativa de fraude, plagio e afins, correspondera em **nota ZERO** para os envolvidos;
- **Códigos ilegíveis serão considerados errados**. A legibilidade é obtida com indentação correta e coerente, bons nomes de variáveis e funções, bem como **boa subdivisão do código** em funções auxiliares;

4. Dicas

- Para **compilar** seu código no terminal:
 - **`gcc lab.c -o lab`**
- **-o** significa *output*. Ele é responsável por gerar o binário do seu programa para execução. É **OBRIGATÓRIO** que o arquivo tenha a função **main**;
- Logo, o que você está dizendo é: “*compile o código **lab.c** com o compilador **gcc**, gerando o executável (saída) **lab**”;*

- Para **executar** seu programa:
 - **`./lab`**

- Você pode baixar os arquivos de casos de teste do run.codes e executá-los manualmente:
 - **`./lab < 01.in`**
- A diretiva **<** redireciona o conteúdo do arquivo *01.in* para o terminal, cujas entradas/dados serão lidas pelo **scanf**;
- Você pode ainda redirecionar a *saída* impressa no terminal para um arquivo:
 - **`./lab < 01.in > 01.res`**

- Por fim, você poder comparar sua resposta com o gabarito (resultado do caso de teste), fazendo
 - **`diff 01.res 01.out`**
 - onde *01.out* é a saída esperada para a entrada *01.in*