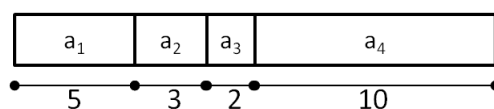


Guerra dos Tronos

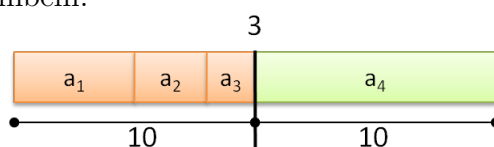
Arquivo-fonte: `guerra.c` ou `guerra.cpp`

Capivarus e Canideus travaram uma guerra durante 50 anos. O motivo da guerra era o tamanho do território de cada reino. Pelo bem da população dos dois reinos, os governantes de seus tronos resolveram fazer um tratado para finalizar a guerra. O tratado consiste em fazer uma divisão justa, e certamente contínua, do território. Eles resolveram pedir sua ajuda para calcular o ponto de divisão do território. Depois de tantos anos de guerra, os reinos não podem lhe pagar uma viagem para ver previamente o território que será dividido. Ao invés disso, eles prepararam uma lista a_1, a_2, \dots, a_N de inteiros que indicam o tamanho de cada seção do território. A seção a_1 é vizinha da seção a_2 que por sua vez é vizinha da seção a_3 ; e assim por diante. Os reinos querem uma divisão em uma seção k de tal forma que $a_1 + a_2 + \dots + a_k = a_{k+1} + a_{k+2} + \dots + a_N$.

Como exemplo, considere o território da figura abaixo, que contém 4 seções, de tamanhos 5, 3, 2 e 10, nesta ordem.



A solução é dividir o território após a seção 3, conforme figura abaixo. Assim, Capivarus fica com 10 de território e Canideus também.



Sua tarefa é, dada uma lista de inteiros positivos a_1, a_2, \dots, a_N , determinar a seção k tal que a soma dos comprimentos das seções a_1 até a_k seja igual à soma dos comprimentos das seções a_{k+1} até a_N .

Entrada

A entrada é composta por duas linhas. A primeira contém um valor inteiro N , que indica o número de seções do território. A segunda linha contém N valores inteiros a_1, a_2, \dots, a_N indicando os comprimentos das seções. Restrições: $1 < N \leq 10^5$; $1 \leq a_i \leq 100, \forall i = 1 \dots N$.

Saída

Seu programa deve imprimir uma única linha contendo um inteiro que indica a seção do território onde acontecerá a divisão.

Observações

É garantido que sempre existe uma divisão que satisfaz as condições dos reinos.

Exemplos

Entrada	Saída
4 5 3 2 10	3
Entrada	Saída
9 2 8 2 8 4 4 4 4 4	4

Obs.: baseada na questão "Guerra por território" da OBI 2012

Número de Euler

Arquivo-fonte: `euler.c` ou `euler.cpp`

O número de Euler, e , que recebeu este nome em homenagem ao matemático suíço Leonhard Euler, é a base dos logaritmos naturais. Ele pode ser definido por:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots$$

onde $n!$ é o fatorial de n , dado por:

$$n! = 1 \times 2 \times \dots \times (n-1) \times n$$

obs.: $0! = 1$.

Faça um programa para calcular uma aproximação do número de Euler com M termos, isto é:

$$e \approx \sum_{n=0}^{M-1} \frac{1}{n!} = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{(M-1)!}$$

Observações

Você pode criar uma função para o cálculo do fatorial.

Entrada

A entrada é composta por um único valor inteiro M , indicando o número de termos a serem utilizados para a aproximação. Restrição: $1 \leq M \leq 10$.

Saída

Seu programa deve imprimir uma linha de saída, contendo um valor real com quatro casas decimais, representando a aproximação obtida para e com M termos.

Exemplos

Entrada	Saída
1	1.0000

Entrada	Saída
5	2.7083

Entrada	Saída
10	2.7183

Fibonacci

Arquivo-fonte: `fibonacci.c` ou `fibonacci.cpp`

A sequência de **Fibonacci** é uma sequência de números naturais cujos dois primeiros elementos são 0 e 1, e todos os elementos subsequentes são calculados pela soma dos dois anteriores.

Matematicamente, representando por $F(n)$ o n -ésimo termo da série, começando de 0, temos que:

- $F(0) = 0$
- $F(1) = 1$
- $F(n) = F(n - 1) + F(n - 2)$, para todo $N \geq 2$.

Assim, os 10 primeiros termos da sequência de Fibonacci são os apresentados a seguir:

n	0	1	2	3	4	5	6	7	8	9
$F(n)$	0	1	1	2	3	5	8	13	21	34

Faça um programa para, dado um número natural n , calcular $F(n)$.

Entrada

A entrada contém um único número natural N . Restrições: $0 \leq N \leq 90$.

Saída

Seu programa deve gerar apenas uma linha de saída, contendo o valor de $F(N)$.

Observações

Preste atenção aos tipos das variáveis, pois $F(90) \approx 2^{61}$.

Exemplos

Entrada	Saída
8	21

Entrada	Saída
15	610

Entrada	Saída
80	23416728348467685

Barragem

Arquivo-fonte: `barragem.c` ou `barragem.cpp`

Na iminência de rompimento de mais uma barragem de rejeitos, certa companhia de mineração está implantando práticas para uma rápida evacuação de pessoas das zonas de risco. Várias simulações estão sendo feitas e você precisa preparar um programa para determinar se cada uma delas foi ou não bem sucedida.

Entrada

A entrada começa com uma linha contendo dois inteiros, P e T , que indicam respectivamente o número de pessoas que deve evacuar o local e o tempo em minutos disponível para isso.

A linha seguinte contém uma sequência de inteiros indicando o número de pessoas evacuadas a cada minuto após soar o alarme da simulação. Ou seja, valores $N_1 N_2 N_3 \dots$ indicando quantas pessoas evacuaram no 1º minuto, no 2º minuto, no 3º minuto e assim por diante. O valor -1 indica o fim da simulação.

Restrições: $P \leq 1000000$, $T \leq 1000$, $0 \leq N_i \leq P$, $N_1 + N_2 + N_3 + \dots \leq P$.

Saída

Seu programa deve escrever apenas uma linha na saída contendo uma das frases seguintes:

- “Todos a salvo”, caso todas as P pessoas sejam evacuadas em até T minutos;
- “Apenas X pessoas a salvo”, caso contrário, sendo X o número de pessoas evacuadas nos primeiros T minutos (em outras palavras, se $X < P$, sendo $X = N_1 + N_2 + \dots + N_T$).

Exemplos

Entrada	Saída
100 5 20 50 30 -1	Todos a salvo
Entrada	Saída
100 5 20 20 20 20 20 -1	Todos a salvo
Entrada	Saída
100 4 20 20 0 40 10 -1	Apenas 80 pessoas a salvo
Entrada	Saída
1000 6 100 200 133 120 120 100 100 100 -1	Apenas 773 pessoas a salvo
Entrada	Saída
50 6 10 10 5 -1	Apenas 25 pessoas a salvo

Número perfeito

Arquivo-fonte: `perfeito.c` ou `perfeito.cpp`

Um número perfeito é um número natural cuja soma de seus divisores, exceto ele mesmo, é igual ao próprio número. Por exemplo, 6 é um número perfeito, pois $1 + 2 + 3 = 6$.

Faça um programa que, dado um número natural N , determine se ele é ou não um número perfeito.

Entrada

A entrada contém apenas um número natural N . Restrição: $1 \leq N \leq 10000$.

Saída

Seu programa deve gerar apenas uma linha na saída, contendo a palavra “SIM” caso N seja um número perfeito ou a palavra “NAO” caso contrário.

Exemplos

Entrada	Saída
6	SIM

Entrada	Saída
28	SIM

Entrada	Saída
360	NAO