

## Super Marco (scrigni)

Limite di tempo: 1.0 secondi  
Limite di memoria: 256 MiB

William sta giocando al suo gioco preferito: Super Marco 64. Per superare un particolare livello all'interno del gioco, è necessario aprire una serie di scrigni nell'ordine corretto.

Il livello è fatto così: ci sono  $n$  scrigni numerati da 1 a  $n$ , ma William non sa quale scrigno corrisponde a quale numero. Se si cerca di aprire uno scrigno nell'ordine sbagliato, Super Marco riceve una scarica elettrica (e lo scrigno non si apre). Gli scrigni aperti fino a quel momento rimangono aperti.

Per esempio, supponiamo che ci siano 3 scrigni numerati così: 3, 1, 2. William, non conoscendo l'ordine, decide di aprire uno scrigno a caso. Per esempio, supponiamo che provi ad aprire quello più a sinistra. Lo scrigno scelto non è il numero 1, quindi non si aprirà e Super Marco prenderà la scossa. Ora William apre lo scrigno di mezzo, che è quello giusto. Prova di nuovo con quello più a sinistra (prendendo di nuovo la scossa), poi prova con quello più a destra (l'ultimo rimasto), e infine apre lo scrigno più a sinistra.

William ha quindi fatto prendere la scossa a Super Marco per ben 2 volte. La domanda però è la seguente: *in media*, quante volte si prende la scossa prima di trovare l'ordine corretto? Ad esempio, nel caso appena citato in cui  $n = 3$ , si può dimostrare facilmente che la scossa si prende in media 1.5 volte.

## Implementazione

Dovrai sottoporre esattamente un file con estensione `.c`, `.cpp` o `.pas`.

🔗 Tra gli allegati a questo task troverai un template (`scrigni.c`, `scrigni.cpp`, `scrigni.pas`) con un esempio di implementazione da completare.

Se sceglierai di utilizzare il template, dovrai implementare la seguente funzione:

C/C++	<code>double scosse(int N);</code>
Pascal	<code>function scosse(N: longint): double;</code>

In cui:

- L'intero  $N$  rappresenta il numero di scrigni.
- La funzione dovrà restituire il numero medio di scosse, che verrà stampato sul file di output.

## Dati di input

Il file `input.txt` è composto da una sola riga che contiene l'unico intero  $N$ .

## Dati di output

Il file `output.txt` è composto da un'unica riga contenente un unico numero reale, la risposta a questo problema.



## Assunzioni

- La risposta verrà considerata corretta se l'errore assoluto o relativo non supererà  $10^{-6}$ .

## Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test relativi ad esso.

- **Subtask 1 [10 punti]:** Casi d'esempio.
- **Subtask 2 [40 punti]:**  $1 \leq N \leq 8$ .
- **Subtask 3 [20 punti]:**  $1 \leq N \leq 12$ .
- **Subtask 4 [20 punti]:**  $1 \leq N \leq 100$ .
- **Subtask 5 [10 punti]:**  $1 \leq N \leq 2\,000\,000\,000$ .

## Esempi di input/output

input.txt	output.txt
3	1.5
4	3