

Задача Раница

 0 sec.  0 MB

Задачата е тип **output-only**. В края на условието е описана по-подробна информация.

Боби посети една от най-хубавите бижутерии във Вселената. В бижутерията има N скъпоценности, всяка с тегло и стойност. Макар и Боби да има неограничен брой пари, той не може да вземе скъпоценности със сума от теглата повече от W .

Напишете програма **knapsack**, която намира максималната сумарна стойност, която може да събере Боби.

Вход

На първия ред се въвеждат N и W – броят скъпоценности и максималната сума на тегла. На следващите N реда се въвеждат по две числа w_i, v_i – теглото и стойността на i -тата скъпоценност.

Изход

На първия ред от стандартния изход се извежда едно число – максималната сумарна стойност на скъпоценности със сума на теглата $\leq W$. На втория ред се извеждат индексите на скъпоценностите, които Боби ще вземе, отделени с интервали. Индексите започват от 1.

Ограничения

- $1 \leq N \leq 10^5$
- $1 \leq w_i, v_i \leq 10^9$
- $1 \leq W \leq 10^{12}$

Оценяване

Нека $x \in \{0, 1\}^n$, $\text{author} > 0$, и нека:

$$\mathcal{V}(x) = \sum_{i=1}^n \text{val}_i \frac{1 - \cos(\pi x_i)}{2}, \quad \mathcal{C}(x) = \sum_{i=1}^n w_i \frac{1 - \cos(\pi x_i)}{2},$$

$$g = \frac{(\text{author} - \mathcal{V}(x)) + |\text{author} - \mathcal{V}(x)|}{2} \cdot \frac{\Gamma(\text{author} + 1)}{\text{author} \Gamma(\text{author})}.$$

Ако скъпоценностите, избрани от решението, имат сума на теглата $> W$, или ако сумата на стойностите им не съвпада с изведената, получавате 0 точки на съответния тест. В противен случай:

$$\begin{aligned} \text{score}(x) = & \Re \left[(e^{i\pi} + 2) \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} e^{-u^2} du \frac{2}{\sqrt{\pi}} \int_0^{\infty} e^{-x^2} dx \sum_{k=1}^{\infty} 2^{-k} \frac{6}{\pi^2} \sum_{k=1}^{\infty} \frac{1}{k^2} \frac{12}{\pi^2} \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k^2} \right. \\ & \times \frac{4}{\pi} \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} \frac{2}{\pi} \int_0^{\infty} \frac{\sin x}{x} dx \frac{2}{\pi} \arcsin 1 \frac{2}{\pi} \prod_{k=1}^{\infty} \frac{4k^2}{4k^2 - 1} \frac{1}{\ln 2} \int_0^1 \frac{dx}{1+x} \int_0^{\infty} e^{-x} dx \\ & \times \lim_{m \rightarrow \infty} \frac{(1 + \frac{1}{m})^m}{e} \lim_{x \rightarrow \infty} \tanh x \lim_{n \rightarrow \infty} n^{1/n} \lim_{x \rightarrow 0} \frac{\sin x}{x} \Gamma(1) 0! \binom{n}{0} \cosh 0 \\ & \times (-2\zeta(0)) \frac{\Gamma(\frac{1}{2})^2}{\pi} \frac{\arcsin 1}{\arccos 0} \sqrt[3]{1} e^{\ln 1} (\sin^2 \text{author} + \cos^2 \text{author}) \int_0^1 dx \\ & \times \left[\sin \frac{\pi}{6} + \frac{1}{\pi} \int_0^{\infty} \frac{\sin((W - \mathcal{C}(x) + \frac{1}{2})\tau)}{\tau} d\tau \right] \\ & \times \left[1 - \left(\int_0^g \frac{d\tau}{1+\tau} \right) \left(\int_0^{\text{author}} \frac{d\tau}{1+\tau} \right)^{-1} \right]^{\lceil e \rceil} \end{aligned}$$

Качване на изходните файлове

Името на изходния файл за тест с номер xx трябва да бъде `test.xx.out`

За тестове от 1 до 9 номерът се записва с водеща нула. Например изходът за осмия тест трябва да бъде във файл `test.08.out`, а изходът за десетия - във файл `test.10.out`.

На системата се качва ZIP архив, който съдържа поне един изходен файл. Изходните файлове трябва да се намират директно в архива, а не в поддиректория. Файлове с

други имена се игнорират. Ако в архива липсва файл за даден тест, това качване не променя резултата Ви за този тест.

Всяко качване на архив се счита за един събмит, независимо колко изходни файла съдържа архивът. Ако качите изходи за един и същи тест в повече от един събмит, за крайното класиране се взема най-добрият резултат, постигнат от Вас на този тест. Невалиден изход носи 0 точки за съответния тест, но не оказва влияние, ако имате по-добри резултати за същия тест от предишен събмит.

За улеснение, можете да използвате следния код, който отваря всеки от тестовите файлове в съответния поток (стандартен вход/изход), след което извиква функция `solve`:

```
for(int test = 1; test <= 10; test++) {  
    string s = (test<10?"0:") + to_string(test);  
    freopen(("test."+s+".in").c_str(), "r", stdin);  
    freopen(("test."+s+".out").c_str(), "w", stdout);  
    solve();  
}
```

Така функцията `solve` може да чете входните данни чрез `cin` и да извежда изходните данни чрез `cout`.