

Задача 5. Fork

Лука отвори Python шел (интерактивен интерпретатор) и написа `os.fork()`, което стартира втори шел. След това, всеки път, когато Лука натисне някой клавиш, той произволно отива към един от двата шела. Всеки шел има входен низ (показан в терминала), който бива редактиран от натисканията на клавиши, отиващи към шела. Освен това Лука вижда терминала и по този начин знае низът на кой шел е засегнат, когато натисне клавиш.

Клавиатурата му има N клавиша с различни знаци върху тях и Backspace. Когато натискането на клавиш със знак отиде в някой шел, знакът просто се добавя към края на неговия входен низ. Когато натискането на клавиша Backspace отиде в шел, последният знак от неговия входен низ се изтрива. Ако входният низ на шела е празен, нищо не се случва с него (въпреки че Лука все пак вижда, че Backspace клавишът е отишъл там). Всяко натискане на клавиш има вероятност P за преминаване към левия шел и вероятност $1 - P$ за преминаване към десния.

Лука иска да въведе някакъв фиксиран низ, състоящ се от N различни знака и в двата шела. Той вече никак не успял да въведе L правилни знака в левия и R в десния. Например, нека разгледаме $P = 0.3$, $N = 2$ (низът може да бъде `ab`), $L = 0$ и $R = 1$. Възможна последователност от събития е:

Стъпка	Клавиш	Страна	Ляв шел	Десен шел
0	-	-	-	a
1	b	Right	-	ab
2	a	Right	-	aba
3	a	Left	a	aba
4	b	Right	a	abab
5	Backspace	Right	a	aba
6	Backspace	Left	-	aba
7	Backspace	Left	-	aba
8	Backspace	Right	-	ab
9	a	Left	a	ab
10	b	Right	a	abb
11	b	Left	ab	abb
12	Backspace	Right	ab	ab

Общо, въвеждането на `ab` в двата шела е отнело 12 натискания на клавиши. Лука се чуди каква би била оптималната му стратегия. По-точно, той иска да знае какъв е минималният очакван (среден) брой натискания на клавиши. Помогнете на Лука, като напишете програма `fork.cpp`, която решава задачата.

Вход

От първия и единствен ред на стандартния вход вашата програма трябва да чете P , N , L и R .

Изход

На първия и единствен ред на стандартния изход вашата програма трябва да изведе изчисления отговор с точност (за предпочитане) 12 цифри или повече. Можете да използвате:

```
std::cout << std::setprecision(12) << ans << std::endl;
```

Ограничения

$$0 \leq L, R \leq N \leq 2 \times 10^7$$

$$0.1 \leq P \leq 0.9$$

Подзадачи и оценяване

Подзадача	Точки	$N \leq$
1	15	5
2	10	15
3	10	35
4	15	100
5	15	450
6	15	1500
7	15	10^6
8	5	2×10^7

За да получите точки за дадена подзадача, Вашето решение трябва успешно да премине всички тестове в нея и във всички предишни подзадачи. За да премине тест, вашето решение трябва да изведе отговор с относителна грешка до 10^{-8} , т.е.:

$$\frac{|yourAns - trueAns|}{trueAns} \leq 10^{-8} \text{ (където } \frac{0}{0} = 0)$$

Примерен тест

Вход	Изход
0.3 2 0 1	16.7142857142857