

1. Last Digit

Zadanie

Wybieramy dodatnią liczbę całkowitą X . Z liczby X wykreślamy ostatnią cyfrę. Postępujemy tak, aż usuniemy wszystkie cyfry liczby X . Następnie sumujemy wszystkie powstałe w ten sposób liczby, włączając liczbę X .

Na przykład, jeżeli wybraliśmy $X = 1234$ to w kolejnych krokach otrzymamy odpowiednio liczby 1234, 123, 12, 1. Ich suma to 1370.

Mamy daną liczbę całkowitą dodatnią S . Proszę napisać program, który znajduje liczbę X taką, że powyżej opisana procedura daje sumę S . Można pokazać, że dla dowolnej dodatniej liczby S istnieje co najwyżej jedna taka wartość X . Jeżeli nie ma takiego X program powinien wypisać -1.

Wejście

W pierwszym i jedynym wierszu standardowego wejścia znajduje się liczba całkowita $1 \leq S \leq 10^{18}$: suma liczb cząstkowych.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać jedną liczbę całkowitą: liczbę X , dającą sumę S lub -1 jeżeli takie X nie istnieje.

Przykład

Dla danych wejściowych:

1370

poprawną odpowiedzią jest:

1234

2. Odd Divisor

Zadanie

Niech $f(x)$ będzie największym nieparzystym dzielnikiem liczby całkowitej dodatniej x . Dana jest dodatnia liczba całkowita N . Napisz program znajdujący $f(1) + f(2) + \dots + f(N)$.

Wejście

W pierwszym i jedynym wierszu standardowego wejścia znajduje się jedna liczba całkowita $1 \leq N \leq 10^9$.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać jedną liczbę całkowitą dodatnią: $f(1) + f(2) + \dots + f(N)$.

Przykład

Dla danych wejściowych:

7

poprawną odpowiedzią jest:

21

Wyjaśnienie:

$$f(1) + f(2) + f(3) + f(4) + f(5) + f(6) + f(7) = 1 + 1 + 3 + 1 + 5 + 3 + 7 = 21.$$

3. Największy substring

Zadanie

Dla dwóch stringów x i y , y jest substringiem x jeżeli y da się uzyskać z x przez usunięcie pewnej liczby znaków (możliwe, że żadnego lub wszystkich). Na przykład, “fnty” jest substringiem “informatyka”, ale “mro” nie jest.

Napisz program, który wyznaczy i wypisze na standardowe wejście leksykograficznie największy substring danego stringu s .

Dla dwóch stringów x i y , x jest leksykograficznie większy niż y jeżeli y jest prefiksem x lub y ma mniejszy znak od x na pierwszej pozycji, na której oba stringi się różnią.

Wejście

W pierwszym i jedynym wierszu standardowego wejścia znajduje się string s , składający się wyłącznie z małych liter alfabetu łacińskiego. Długość stringu należy do przedziału $[1, 50]$.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać string będący leksykograficznie największym substringiem stringu s .

Przykład

Dla danych wejściowych:

```
test
```

poprawną odpowiedzią jest:

```
tt
```

Wszystkie substringi stringu “test” (w kolejności leksykograficznej) to:

“”, “e”, “es”, “est”, “et”, “s”, “st”, “t”, “te”, “tes”, “test”, “tet”, “ts”, “tst” i “tt”.

“tt” jest więc leksykograficznie największym substringiem stringu s .