Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie
Ogólnopolska Olimpiada
"O Diamentowy Indeks AGH" 2020/21

Informatyka – Etap III

Maksymalna ocena za każde zadanie to 20 punktów.

Zadanie 1: Last Digit

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
```

Zadanie 2: Odd Divisor

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

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: $f(1) + f(2) + \ldots + f(N)$.

Przykład

Dla danych wejściowych:

```
7
```

poprawną odpowiedzią jest:

```
21
```

Wyjaśnienie:

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

Zadanie 3: Największy substring

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, "ompa" jest substringiem "olimpiada", ale "dp" 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.

Zadanie 4: Ring

Dany jest string s, stanowiący okres nieskończonego periodycznego stringu t. Na przykład jeżeli $s =$ "abc" to $t =$ "abcabcabc...". Niech n będzie długością s. Tworzymy nowy string o długości n w sposób następujący: wybieramy przesunięcie $o >= 0$ i krok $p < n$, będący liczbą pierwszą. Nowy string składa się z pierwszych n znaków jakie możemy przeczytać ze stringu t zaczynając od indeksu o i następnie przesuwając się o p pozycji w prawo.

Formalnie, nowy string będzie się składał z następujących znaków (w tym porządku): $t[o]$, $t[o+p]$, $t[o+2p], \ldots, t[o+(n-1)p]$. Znajdź i wypisz na standardowe wyjście najmniejszy leksykograficznie string, jaki można w ten sposób uzyskać.

Dla danych dwóch różnych stringów, mniejszy leksykograficznie jest ten, który zawiera mniejszy znak na pierwszej pozycji, na której stringi się różnią.

Liczba 1 **nie** jest liczbą pierwszą.

Wejście

W pierwszym i jedynym wierszu standardowego wejścia znajduje się string s o długości $3 \leq |s| \leq 50$ składający się wyłącznie z małych liter alfabetu łacińskiego.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać najmniejszy leksykograficznie string o długości $|s|$, który da się otrzymać stosując opisaną wyżej procedurę.

Przykład

Dla danych wejściowych:

```
cba
```

poprawną odpowiedzią jest:

```
abc
```

Wybieramy przesunięcie $o = 2$ i krok $p = 2$ i uzyskujemy nowy string: $t[2] + t[4] + t[6] =$ 'a'+'b'+'c' = "abc".

Zadanie 5: The OR Game

Dana jest liczba docelowa G oraz tablica dodatnich, unikalnych liczb całkowitych $T[N]$. Zaczynamy od liczby $X = 0$. Zadaniem gry jest uzyskanie liczby G w jednym lub więcej krokach. W każdym kroku wybieramy dowolną liczbę z tablicy T i zastępujemy X przez alternatywę bitową X i wybranego elementu T.

Napisz program, który wyznaczy i wypisze na standardowe wyjście minimalną liczbę elementów tablicy T, które należy z niej usunąć aby nie dało się uzyskać liczby G.

Jeżeli a i b są pojedynczymi bitami ich alternatywa bitowa $a|b = \max(a, b)$. Alternatywą bitową dwóch liczb całkowitych, A i B, o reprezentacjach bitowych odpowiednio $A = a_n \dots a_1$ i $b_n \dots b_1$ jest liczba $C = A|B = c_n \dots c_1$, gdzie $c_i = a_i|b_i$. Na przykład $10|3 = (1010)_2|(0011)_2 = (1011)_2 = 11$.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite $1 \leq N \leq 20$: długość tablicy T i $1 \leq G \leq 10^9$: liczba docelowa. Kolejny wiersz zawiera dokładnie N liczb z przedziału $[1, 10^9]$: elementy tablicy T. Dla $i \neq j : T[i] \neq T[j]$.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia program powinien wypisać jedną liczbę całkowitą: liczbę elementów tablicy T, którą należy usunąć by nie dało się uzyskać liczby docelowej G.

Przykład

Dla danych wejściowych:

```
5 7
1 2 4 7 8
```

poprawną odpowiedzią jest:

```
2
```

W tym przykładzie należy usunąć liczbę 7 i jedną z liczb 1, 2, 4.