

Obnova hesla

Bežní používatelia používajú ľahké heslá (typu "heslo", "qwerty", "nbusr123" apod.), a tak sa správcovia jedného nemenovaného informačného systému rozhodli ako heslo zvoliť permutáciu prvých n kladných celých čísel.

Pri nastavovaní hesla sa kontroluje, či sa jedná o platné heslo. Teda, či je tvorené n rôznymi číslami z rozsahu 1 až n . Programátori to kontrolujú usporiadaním čísel použitím algoritmu usporiadania spájaním (MergeSort) a následnou kontrolou s aritmetickou postupnosťou $1, 2, 3, \dots, n$.

Podarilo sa nám zistiť, že pri takomto usporiadaní používajú nižšie uvedený kód (napísaný v Pythone 3) aj s výstupom na ladenie (označené ako debug). Dokážte, že tento výstup môže byť použitý ako zadné vrátka, t.j. zrekonštruujte permutáciu zo zadaného výstupu.

Úloha

Pre zadaný výstup funkcie `merge_sort` určte pôvodnú permutáciu na vstupe.

```
def merge_sort(arr) :
    n = len(arr)
    if n <= 1 :
        return arr
    mid = n//2
    return merge(merge_sort(arr[:mid]), merge_sort(arr[mid:]))

def merge(arr1, arr2) :
    result = []
    while (len(arr1) > 0) and (len(arr2) > 0) :
        if arr1[0] < arr2[0] :
            print('+ ', end='') # debug
            result.append(arr1[0])
            arr1 = arr1[1:]
        else:
            print('- ', end='') # debug
            result.append(arr2[0])
            arr2 = arr2[1:]

    result.extend(arr1)
    result.extend(arr2)
    return result
```

Tento kód môžete spustiť v Pythone verzie 3, napr. [online](#) alebo aj s [vizualizáciou volaní](#).

Vstup

Vstup obsahuje dva riadky. Prvý riadok obsahuje jedno kladné celé číslo, $2 \leq N \leq 10.000$, počet čísel vo vstupnej permutácii.

Druhý riadok obsahuje výstup uvedenej funkcie - postupnosť znakov '+' a '-'. Tento reťazec je neprázdny a nie je dlhší ako 200.000 znakov.

Výstup

Na jediný riadok výstupu vypíšte hľadanú permutáciu. Medzi každými dvoma číslami vypíšte jednu medzeru.

Môžete predpokladať, že naozaj existuje korektná permutácia dĺžky N taká, že kontrolný výstup jej spracovania programom vyššie ste dostali na vstupe. Môžete predpokladať, že výstup je jednoznačný.

Príklady

Vstup:

2
+

Výstup:

1 2

Vstup:

2
-

Výstup:

2 1

Vstup:

4
+--+ -

Výstup:

2 4 3 1

Vstup:

10
+---+---+---+---+---+---+---+---+---+---

Výstup:

1 6 10 8 2 4 9 5 3 7