

algoritmy a dátové štruktúry, ZS 2023/24

druhá domáca úloha, termín do 27. 10. 2023

V pamäti je globálne pole a a chceme ho utriediť pomocou nasledovnej funkcie `mergeSort`:

```
void mergeSort(int a[], int l, int r) {
    kvik();
    if (sorted(a, l, r)) return;
    int mid = (l + r) / 2;
    mergeSort(a, l, mid);
    mergeSort(a, mid, r);
    merge(a, l, mid, r);
}
```

Funkcia `sorted(a, l, r)` vráti `true`, ak je interval $[l, r)$ v poli a utriedený, t.j. ak pre každé $l \leq i < r-1$ platí $a[i] \leq a[i+1]$. Funkcia `merge(a, l, mid, r)` spojí utriedené intervaly $[l, mid)$ a $[mid, r)$ z poľa a a výsledok uloží do intervalu $[l, r)$ poľa a . Funkcia `kvik()` nahlas zakvíka. Pole a na začiatku obsahuje n prvkov $a[0], a[1], \dots, a[n-1]$, takže na jeho utriedenie sa zavolá `mergeSort(a, 0, n)`.

Máte za úlohu napísať program, ktorý pre zadané n a k nájde permutáciu prvkov $\{1, 2, \dots, n\}$ tak, že keď ju budeme triediť pomocou funkcie `mergeSort`, tak počas triedenia bude počet k kvíknutí.

Vstup

Na vstupe sú dve celé čísla n a k , $1 \leq n \leq 10^6$, $1 \leq k \leq 2 \cdot 10^6$.

Výstup

Vypíšte jeden riadok, kde bude medzerou oddelených n čísel tvoriacich permutáciu $\{1, 2, \dots, n\}$ (t.j. každé z čísel $1, \dots, n$ sa na výstupe nachádza práve raz) takú, že volanie `mergeSort` na nej k -krát zakvíka. Ak taká permutácia neexistuje, vypíšte namiesto toho -1 . Ak existuje viacero permutácií, vypíšte ľubovoľnú z nich.

Príklady

5 3

2 5 1 3 4

Na vstupe 2 5 1 3 4 sa najprv zavolá `mergeSort(a, 0, 5)`, ktorý kvíkne raz. Zistí sa, že celé pole (interval $[0, 5)$) nie je utriedené. Nastaví sa `mid=2` a zavolá sa `mergeSort(a, 0, 2)`. Toto volanie kvíkne, zistí, že interval $[0, 2)$ je utriedený (obsahuje prvky 2 5) a skončí. Program sa vráti do prvého volania a zavolá sa `mergeSort(a, 2, 5)`. Ten kvíkne, zistí, že interval $[2, 5)$ (s prvkami 1 3 4) je utriedený a skončí. Prvé volanie nakoniec spojí oba úseky do utriedeného poľa a skončí. Celkovo teda boli 3 kvíknutia.

6 11

6 5 4 3 2 1

47 42

-1

11 5

5 11 1 4 8 2 3 6 7 9 10