

Egyengető

Adott pozitív egész számoknak az $A = (a_1, \dots, a_N)$ sorozata és egy K pozitív egész szám. Azt mondjuk, hogy az A sorozat K -kiegyenlített, ha bármely két elemének az eltérése legfeljebb K , azaz minden $1 \leq i, j \leq N$ indexpárra $|a_i - a_j| \leq K$ teljesül.

Minden A sorozat K -kiegyenlítetté tehető oly módon, hogy minden a_i eleméhez hozzáadunk egy-egy alkalmas d_i egész számot. Egy ilyen kiegyengetés költsége

$$\sum |d_i| = |d_1| + |d_2| + \dots + |d_N|.$$

Írj programot, amely megadja az A sorozat K -kiegyenlítésének a minimális költségét!

A feladatot interaktív módon kell megoldani! A sorozat elemeit egyesével beolvasva, minden elemet követően meg kell adni az addig beolvasott sorozatelemek K -kiegyenlítésének a minimális költségét!

Könyvtár

A programodnak a következő könyvtári műveleteket kell használnia.

- `int getN()`: a program elején kell egyszer hívni, a sorozat N elemszámát adja ($1 \leq N \leq 200\,000$).
- `int getK()`: a `getN` után kell egyszer hívni és a K értékét adja ($0 \leq K \leq 10^8$).
- `int Data()`: minden híváskor a sorozat következő elemét adja ($0 < a_i \leq 10^8$). Csak a `getK` hívása után válik elérhetővé, és addig nem használható újra, amíg a `Solution` művelet nem került meghívásra.
- `Solution(long long m)`: minden `Data()` hívás után ezzel kell közölni az addig megkapott sorozat K -kiegyenlítésének minimális m költségét.

A műveletek használatához a programodnak tartalmaznia kell az `#include "grader.h"` importáló sort!

Gyakorlás

A letölthető `minta.zip` egy `minta "grader"` modult tartalmaz. Add hozzá a fejlesztői környezetedben a megoldásod projektjéhez a `grader.h` és a `grader.cpp` állományokat. A megoldásodat tartalmazó `main.cpp` állomány is legyen ott a projektben. Ezt követően elérhetővé válnak a programod számára a fenti könyvtári műveletek.

A modul a program futása során a standard bemenetről olvassa be az adatokat (a `getN()` művelet első hívásakor). Az első sor az N és a K értékét tartalmazza, a második sor tartalmazza az A sorozat elemeit. A `be1.txt` és `be2.txt` ennek megfelelő formájú példa bemenetek.

A `"Solution"` függvény hívása nem ellenőrzi a megadott m érték helyességét! A modul csak a könyvtári műveletek helyes alkalmazását vizsgálja és a standard kimeneten jelzi az ezzel kapcsolatos hibákat. A példa bemenetekre a helyes megoldási értékeket a `ki1.txt` és `ki2.txt` fájlok tartalmazzák.

Csak az általad készített `main.cpp` állományt kell beadnod!

Korlátok

Időlimit: 1.6 mp.

Memórialimit: 32 MB

Pontozás

Részfeladat	Korlátok	Pontszám
1	a minta	0
2	$K=0$	20
3	$N \leq 1000$ és $a_i \leq 1\,000\,000$	20
4	$N \leq 10\,000$ és $a_i \leq 1\,000\,000$	20
5	nincsenek további korlátok	40