Problem A. Bástya

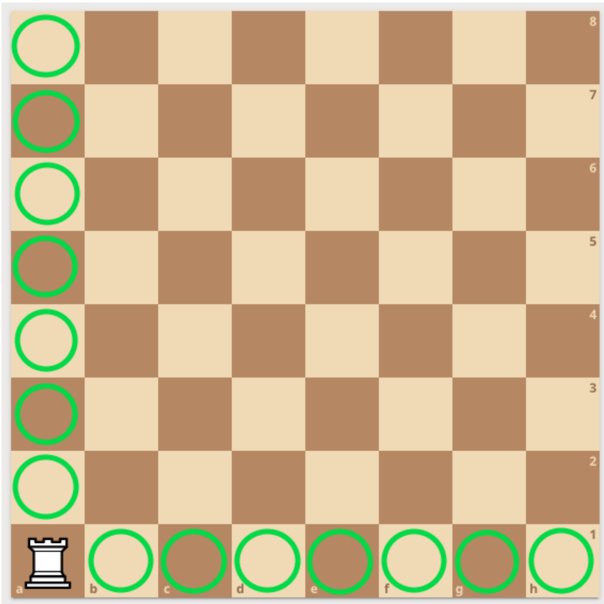
Time limit: 1 second
Memory limit: 256 megabytes

Adott egy $n \times m$ méretű sakktábla. Vagyis n sorral és m oszloppal.

Ezen a sakktáblán csak egy figura van — a bástya. Ez a bal alsó sarokban található. Nincsenek más figurák.

Emlékezzünk, hogy a bástya bármennyi pozitív számú négyzetet tud mozogni vízszintesen vagy függőlegesen egy lépésben, de nem átlósan.

Határozd meg, hány négyzetre tud a bástya pontosan egy lépésben eljutni.



A képen egy hagyományos 8×8 méretű sakktábla látható. Ezen a bástya az összes zölddel jelölt négyzetre tud lépni. Összesen 14 ilyen négyzet van, tehát a válasz 14.

Input

Az első sor egy egész számot tartalmaz n ($1 \leq n \leq 20$).

A második sor egy egész számot tartalmaz m ($1 \leq m \leq 20$).

Output

Írd ki a négyzetek számát, ahová a bástya egy lépésben el tud jutni.

Examples

standard input	standard output
8 8	14
3 2	3

Note

A képen látható magyarázat alapján az első példára a válasz 14.

A második példában a válasz 3, mert a bástya csak egy pozíciót tud jobbra és két pozíciót felfelé lépni.

Problem B. Koordináták

Time limit: 1 second
Memory limit: 256 megabytes

Adott egy pont (x, y, z) a 3D térben.

Meg kell találni a **négyzetes** távolságot ettől a ponttól az origóig (azaz a $(0, 0, 0)$ pontig).

Emlékezzünk, hogy a távolság két pont között (x_1, y_1, z_1) és (x_2, y_2, z_2) a következő képlettel van meghatározva:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Input

Az első sor egy egész számot tartalmaz x $(-100 \le x \le 100)$.

A második sor egy egész számot tartalmaz y $(-100 \le y \le 100)$.

A harmadik sor egy egész számot tartalmaz z $(-100 \le z \le 100)$.

Output

Adjon meg egy egész számot.

Example

standard input	standard output
1 -3 5	35

Note

Az első tesztben a $(1, -3, 5)$ pont és a $(0, 0, 0)$ pont közötti négyzetes távolság érdekel minket.

A koordináták behelyettesítésével a képletbe a következőt kapjuk:

$$\left(\sqrt{(1-0)^2 + (-3-0)^2 + (5-0)^2}\right)^2 =$$

$$= \left(\sqrt{1 + 9 + 25}\right)^2 = 35$$

Problem C. Készítsd el a Házi Feladatot

Time limit: 1 second
Memory limit: 256 megabytes

Sakurako a következő programozási órájára készül, amely ma $14:00$-kor kezdődik, de be kell fejeznie a befejezetlen házi feladatát. Az óra a perc múlva kezdődik, és a házi feladat befejezése b percet vesz igénybe. Szerencsére Sakurako tegnap c percet dolgozott a házi feladaton.

A feladatod az, hogy meghatározd, vajon Sakurako be tudja-e fejezni a házi feladatát az óra kezdete előtt.

Például, ha az óra 5 perc múlva kezdődik, Sakurako házi feladata 6 percet igényel, és ő már 2 percet dolgozott rajta. Ebben az esetben $6 - 2 = 4$ perc szükséges a befejezéshez. Ezért, ha azonnal elkezdi az órára való felkészülést, befejezi a házi feladatát (sőt, még egy perc szabadideje is marad).

Input

Egyetlen sor három egész számot tartalmaz: a, b és c ($0 \le a, b, c \le 360$).

Output

Írd ki "`IGEN`", ha Sakurakónak van esélye befejezni a házi feladatát az óra kezdete előtt, vagy "`NEM`" egyébként.

Scoring

Bizonyos számú pontot kapsz, ha a megoldásod helyesen működik $c = 0$ esetén; azaz Sakurako nem dolgozott a házi feladatán tegnap.

Examples

standard input	standard output
5 6 2	YES
5 8 3	YES
5 8 1	NO
0 0 0	YES

Note

Az első példát a legendában magyarázták el.

A második példában az óra 5 perc múlva kezdődik. Sakurakónak $8 - 3 = 5$ percre van szüksége a házi feladat befejezéséhez. Ezért tökéletesen belefér az időbeosztásába.

A negyedik példában az óra már elkezdődött, de a házi feladat befejezése nulla percet igényel (vajon szóbeli házi feladat volt?).

Problem D. Labdák és Kukák

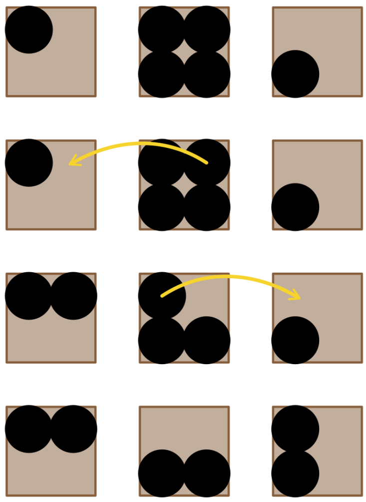
Time limit: 1 second
Memory limit: 256 megabytes

Sakurako egy fiatal programozó, aki egy robotikai projekten dolgozik.

Három okos kukát épített, amelyek kezdetben a, b és c labdával rendelkeznek. A kukák úgy vannak tervezve, hogy labdákat adjanak át egymásnak, és az ő feladata egy olyan program megírása, amely egyenlővé teszi a labdák számát mindhárom kukában.

Sakurako utasíthat egy nem üres kukát, hogy vegyen ki egy labdát és adja át egy másiknak. Azonban kíváncsi arra, hogy lehetséges-e egyáltalán kiegyenlíteni a kukákat, és ha igen, hány lépésbe telik ez. Tudsz segíteni neki egy hatékony megoldás megtervezésében?

Tegyük fel, hogy három kukája van; az első kukában 1 labda van, a második kukában 4 labda van, a harmadik kukában pedig 1 labda van. Az alábbiakban található egy illusztráció egy ilyen példához.



A sárga nyilak mutatják, hogy melyik labdát vesszük ki és hova tesszük. Ebben az esetben két műveletet hajtunk végre.

Input

Az egyetlen sor három egész számot tartalmaz: a, b és c ($0 \le a, b, c \le 10^8$).

Output

Ha lehetetlen egyenlővé tenni a labdák számát mindhárom kukában, akkor írd ki "-1".

Ellenkező esetben írd ki a labdák egyenlővé tételéhez szükséges minimális lépések számát.

Scoring

Legalább 25 pontot kapsz, ha a megoldásod helyesen működik, amikor $b = 0$ és $c = 0$; azaz a második és harmadik kuka üres.

Legalább 25 pontot kapsz, ha a megoldásod helyesen működik, amikor $b = 2 \cdot a$ és $c = 3 \cdot a$.

Examples

standard input	standard output
1 4 1	2
2 2 3	-1
0 0 0	0
12 0 0	8

Note

A második példában megmutatható, hogy lehetetlen a labdákat a kukák között elosztani a leírtak szerint.

A harmadik példában az összes kuka üres, ami azt jelenti, hogy mindegyiknek ugyanannyi labdája van; ezért a kimenet nulla.

A negyedik példában a következő stratégiát követhetjük:

$$\begin{array}{lcl} (12,0,0) & \to & (11,1,0) \\ (11,1,0) & \to & (10,1,1) \\ (10,1,1) & \to & (9,2,1) \\ (9,2,1) & \to & (8,2,2) \\ (8,2,2) & \to & (7,3,2) \\ (7,3,2) & \to & (6,4,2) \\ (6,4,2) & \to & (5,4,3) \\ (5,4,3) & \to & (4,4,4) \end{array}$$

ami pontosan 8 műveletet igényel. Bizonyítani tudjuk, hogy lehetetlen a kukákat kevesebb mint 8 művelettel kiegyenlíteni.

Problem E. Sakurako Késik

Time limit: 1 second
Memory limit: 256 megabytes

Újabb nap az iskolában: újabb alkalom, amikor Sakurako késik!

Ma elaludt, és a lehető leggyorsabban el kell jutnia az iskolába.

Van n gyalogos átkelő, amely Sakurakót elválasztja az iskolájától; mindegyik egyetlen közlekedési lámpát tartalmaz. Minden közlekedési lámpa vagy zöld, vagy piros. Minden egyes közlekedési lámpa színe minden percben változik.

Sakurako nem igazán gyors, ezért egy percet tölt el egy gyalogos átkelő átkelésével. Emellett nagyon törvénytisztelő, ezért nem lépi át az átkelőt, ha a lámpa piros. Végül, ha a lámpa zöld, akkor **mindig** átkel az úton.

Határozd meg, mennyi a minimális idő, amit Sakurakónak el kell töltenie ahhoz, hogy eljusson az iskolába.

Input

Az első sorban egy egész szám n ($1 \le n \le 10^6$) található, amely a gyalogos átkelők számát jelöli, amelyeket Sakurakónak át kell lépnie.

A második sorban egy n karakterből álló s karakterlánc található, amely minden közlekedési lámpa kezdeti színét jelöli; minden karakter vagy "`G`" (zöld), vagy "`R`" (piros). Az első karakter a legközelebbi közlekedési lámpa színét jelöli, az utolsó karakter pedig a legmesszebbi közlekedési lámpa színét.

Output

Egyetlen sorban egy egész számot kell kiírni, amely a Sakurakónak az átkelők átlépéséhez szükséges minimális időt jelöli.

Scoring

Legalább 30 pontot kapsz, ha a megoldásod helyesen működik, amikor $s_1 = s_2 = \cdots = s_{n-1} = s_n$; azaz, minden szín azonos.

Legalább 30 pontot kapsz, ha a megoldásod helyesen működik, amikor $s_i \ne s_{i+1}$ ($1 \le i \le n-1$); azaz, minden szomszédos szín különböző.

Examples

standard input	standard output
5 RGRRG	7
10 GGRRGRGRRG	13

Note

Az első példában Sakurako a következő módon haladhat:

Idő 0:

Sakurako a 0-ás pozícióban van.

Idő 1:

Sakurako a 0-ás pozícióban van, és elkezd átkelni az 1-es gyalogos átkelőn.

Idő 2:

Sakurako az 1-es pozícióban van, és elkezd átkelni a 2-es gyalogos átkelőn.

Idő 3:

Sakurako a 2-es pozícióban van, és elkezd átkelni a 3-as gyalogos átkelőn.

Idő 4:

Sakurako a 3-as pozícióban van.

Idő 5:

Sakurako a 3-as pozícióban van, és elkezd átkelni a 4-es gyalogos átkelőn.

Idő 6:

Sakurako a 4-es pozícióban van, és elkezd átkelni az 5-ös gyalogos átkelőn.

Idő 7:

Sakurako átkelt az összes 5 átkelőn.

A második példában megmutathatjuk, hogy a minimális idő, amelyre Sakurakónak szüksége van az összes átkelő átlépéséhez, 13.

Problem F. Gyönyörű String

Time limit: 1 second
Memory limit: 256 megabytes

A bináris string olyan string, amely csak "0"-t és "1"-t tartalmaz.

A string egy részstringje olyan string, amelyet úgy lehet kapni, hogy eltávolítunk néhány karaktert a kezdetről és/vagy a végéről.

Sakurako gyönyörűnek tart egy bináris stringet, ha az egyesek száma legalább 2, és ez a szám osztja a string hosszát.

Például a "1001", "011100" és "010100100" gyönyörű stringek, de a "101", "01010" és "1001000" nem gyönyörűek.

Sakurako szereti a gyönyörű stringeket, de rossz a keresésükben. Ezért megkér, hogy segíts neki találni bármilyen gyönyörű részstringet a megadott stringből.

Input

Az első sor tartalmaz egy egész számot n ($3 \le n \le 2 \cdot 10^5$) — a string hossza.

A második sor tartalmaz egy bináris stringet — a stringet, ahol Sakurako elvesztette a gyönyörű stringjét.

Garantált, hogy legalább három "1" karakter található a stringben.

Bizonyítani lehet, hogy a válasz mindig létezik.

Output

Adjon meg két egész számot l és r ($1 \le l \le r \le n$), ahol a l-től kezdődő és az r-rel végződő részstring gyönyörű. Ha több megoldás is van, adjon meg bármelyiket.

Scoring

Legalább 32 pontot kap, ha a megoldása helyesen működik $n \le 1000$ esetén.

Examples

standard input	standard output
5 10101	1 4
6 101101	3 4

Note

Az első példában a "1010" részstring gyönyörű, mivel a hossza, ami 4, osztható az egyesek számával, ami 2.

A második példában a "11" részstring gyönyörű, mivel a hossza, ami 2, osztható az egyesek számával, ami 2.

Problem G. Szép Tömb

Time limit: 1 second
Memory limit: 256 megabytes

A a tömb szép, ha minden $a_i - a_{i-1}$ $(2 \le i \le n)$ különböző.

Például, a $[1, 3, 2, 9]$ és a $[8, 4, 6, 9]$ szép, míg a $[1, 2, 3, 9]$ és a $[1, 5, 5, 9]$ nem az.

Egy $b_1, b_2, \ldots, b_n$ egész számokból álló sorozat permutációnak nevezhető a $c_1, c_2, \ldots, c_n$ sorozatra, ha ezekben a tömbökben minden egyedi szám száma megegyezik. Például, a $[1, 5, 3, 4, 2]$ egy permutációja a $[5, 2, 1, 4, 3]$-nak, és a $[1, 3, 3, 5]$ egy permutációja a $[3, 5, 1, 3]$-nak. Azonban a $[1, 3, 4]$ nem permutációja a $[2, 3, 4]$-nek (mivel nem tartalmazza a 2-t, amelynek szerepelnie kellene a sorozatban, és tartalmazza az 1-et, amelynek nem kellene szerepelnie).

Sakurako kapott egy tömböt. Ő egy példát szeretne találni a megadott tömb permutációjára, amely szép.

Input

Az első sor egy egész számot tartalmaz n $(2 \le n \le 2 \cdot 10^5)$ — a tömb hossza.

A második sor n egész számot tartalmaz $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^9)$ — a tömb elemei.

Garantált, hogy a tömbben minden szám különböző. Azt is garantálják, hogy létezik megoldás.

Output

Adjon meg egy szép tömböt (amely az eredeti tömb permutációja) egy sorban. Ha több megoldás is létezik, adjon meg bármelyiket.

Scoring

Legalább 16 pontot kap, ha a megoldása helyesen működik $n \le 10$ esetén.

Legalább 48 pontot kap, ha a megoldása helyesen működik $n \le 1000$ esetén.

Examples

standard input	standard output
5 1 2 3 4 5	3 5 4 1 2
3 4 2 3	4 2 3

Note

Az első példában a $[3, 5, 4, 1, 2]$ tömb szép, mivel ha az $a_i - a_{i-1}$ értékeket vesszük minden i-re, akkor a következőket kapjuk:

- $5 - 3 = 2$;
- $4 - 5 = -1$;
- $1 - 4 = -3$;
- $2 - 1 = 1$.

Ahogy látjuk, minden szám különböző.

A második példában a $[4, 2, 3]$ tömb szép, mivel ha az $a_i - a_{i-1}$ értékeket vesszük minden i-re, akkor a következőket kapjuk:

- $2 - 4 = -2$;
- $3 - 2 = 1$.

Ahogy látjuk, minden szám különböző.

Problem H. Sakurako és egy Karácsonyi Tömb

Time limit:	1.5 seconds
Memory limit:	256 megabytes

Sakurako nemrégiben egy tömböt díszített fel úgy, hogy az Karácsonyi tömbbé vált. Ez **fényes** volt, amíg Chefir meg nem zavarta.

Egy n hosszúságú tömb **fényes**-nek nevezhető, ha a következő érvényes:

- minden i esetén $(1 \le i \le n)$ $a_i = a_{n-i+1}$;
- és van egy elem, amely legalább $\frac{n}{2}$ alkalommal előfordul.

Adott egy a tömb, amelynek hossza n, ahol n páros. Meg kell határoznod, hogy hány elemet kell minimálisan kicserélni ahhoz, hogy ez a tömb **fényes** legyen.

A "kicserél"azt jelenti, hogy választhatsz bármely i-t $(1 \le i \le n)$, és hozzárendelheted $a_i = x$-et, ahol x egy tetszőleges egész szám (akkor is, ha x nem létezik a tömbben).

Input

Az első sorban egy egész szám n $(2 \le n \le 5 \cdot 10^5$; n páros), amely a tömb hosszát jelöli.

A második sorban n egész szám található: $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^6)$.

Output

Egy sorban adj meg egy egész számot, amely a minimális elemek számát jelöli, amelyeket ki kell cserélni ahhoz, hogy a **fényes** tömbbé váljon.

Scoring

Legalább 10 pontot kapsz, ha a megoldásod helyesen működik, ha $a_i = a_{i+1}$ minden $(1 \le i < n)$ esetén.

Legalább 20 pontot kapsz, ha a megoldásod helyesen működik, ha $a_i = a_{n-i+1}$ minden i esetén $(1 \le i \le n)$.

Legalább 30 pontot kapsz, ha a megoldásod helyesen működik, ha $n \le 1000$.

További 40 pontot kapsz, ha a megoldásod helyesen működik korlátozások nélkül.

Examples

standard input	standard output
6 2 6 4 6 2 9	3
6 3 7 2 4 9 3	3

Note

Optimális csere az 1. tesztesetben:

$[2, 6, 4, 6, 2, 9] \to [2, 2, 6, 6, 2, 2]$ Ezen tömb esetén mindkét feltétel teljesül:

- minden $i\,(1 \le i \le n)$ esetén $a_i = a_{n-i+1}$
- a 2 egész szám több mint $\frac{6}{2} = 3$ alkalommal előfordul a tömbünkben.

Meg tudjuk mutatni, hogy ez a minimális műveletek száma, amit végre kell hajtanunk.

A 2. tesztesetben az optimális csere:

$[3, 7, 2, 4, 9, 3] \rightarrow [3, 4, 4, 4, 4, 3]$

Meg lehet mutatni, hogy ez a legkisebb műveletek száma, amit végre kell hajtanunk.