Problème A: Allumettes.

Temps limite : 1s

Brieuc et Aymeric se rendent au KARWa, mais ils s'ennuient dans le train. Pour passer le temps, ils jouent aux allumettes. Le jeu consiste en n allumettes, et à chaque tour, chaque joueur peut enlever de 1 à k allumettes. **Le joueur qui enlève la dernière allumette a gagné**.

Brieuc est un petit malin, et il préfère donc commencer. Pouvez-vous déterminer qui va gagner la partie si les deux joueurs jouent de manière optimale ?

Input :

Deux entiers n ($2 \leq n \leq 10^9$) et k ($1 \leq k < n$), où n et k représentent respectivement le nombre total d'allumettes et le nombre d'allumettes que chaque joueur peut enlever à chaque tour.

Output :

Le nom du gagnant du jeu si les deux joueurs jouent de manière optimale : "`Brieuc`" si c'est le premier joueur qui gagne, "`Aymeric`" sinon.

Exemple d'input et son output correspondant :

Input 1	Output 1
`4 3`	`Aymeric`

Input 2	Output 2
`10 3`	`Brieuc`

Problème B: Bonheur.

Temps limite : 2s

Alexis recherche deux coéquipiers pour participer au KARWa. Il dispose d'une liste de n personnes sans équipe, triées par un identifiant unique. Alexis possède un nombre porte-bonheur k et il souhaite savoir s'il est possible de trouver deux coéquipiers dont la somme des identifiants est égale à k.

Input :

L'entrée consiste en :

- une ligne avec deux entiers : n ($2 \leq n \leq 10^6$) représentant le nombre de personnes seules, et k ($2 \leq k \leq 10^9$) représentant le nombre porte-bonheur d'Alexis.
- une seconde ligne contenant n entiers $x_1, x_2, \ldots, x_n$ ($1 \leq x_i < k$) en ordre croissant, représentant les identifiants des personnes sans équipe.

Output :

Si Alexis peut trouver deux coéquipiers dont la somme des identifiants est égale à k, le programme doit afficher "`yes`", Sinon, il doit afficher "`no`".

Exemple d'input et son output correspondant :

Input 1

```
10 69
1 3 4 7 11 13 21 27 42 69
```

Output 1

```
yes
```

Input 2

```
5 12
1 3 7 8 10
```

Output 2

```
no
```

Problème C: Circuit Confort.

Temps limite : 1s

Dans le cadre de la campagne *Circuit Confort*, Bertrand Karwa et son équipe de Sécurail sont réhabilités temporairement pour gérer la sécurité de plusieurs lignes ferroviaires belges. Ils reçoivent une liste de lignes entre deux gares à emprunter, et ils doivent passer au moins une fois sur chaque ligne afin de montrer aux voyageurs leur présence et ainsi les rassurer.

Ils souhaitent optimiser leur chemin dans le but d'emprunter chaque ligne de train exactement une fois, indépendamment du sens dans lequel il la prenne, mais ils ne sont pas sûrs de la faisabilité de la tâche. Aidez-les à déterminer s'il leur est possible **en démarrant à une gare donnée**, de suivre un chemin pour **rentrer à cette même gare** et en passant **une et une seule fois** sur chaque ligne de train.

Input :

L'entrée consiste en :

- une ligne avec un entier n ($3 \leq n \leq 10^3$), le nombre de chemins que Mr. Karwa et son équipe doivent emprunter,
- une ligne avec un nom de gare, là d'où doivent partir Mr. Karwa et son équipe,
- n lignes contenant un chemin à emprunter entre deux gares, c'est-à-dire deux chaînes de caractères séparées par un espace "`gare1 gare2`" pour dire qu'ils doivent emprunter soit la ligne de "`gare1`" à "`gare2`" ou soit de "`gare2`" à "`gare1`" (mais pas les deux).

Les noms de gares sont donnés en lettre alphabétique minuscule. Une ligne de train entre deux gares ne sera donnée qu'une seule fois et agit comme une ligne à double sens.

Output :

S'il est impossible pour Mr. Karwa et son équipe de visiter exactement une fois chaque ligne de train donnée, donnez en sortie "`impossible`", sinon donnez "`ok`".

Exemple d'input et son output correspondant :

Input 1

```
6
mons
mons louvainlaneuve
louvainlaneuve bruxelles
bruxelles anvers
anvers louvainlaneuve
louvainlaneuve malines
malines mons
```

Output 1

```
ok
```

Input 2

```
7
louvainlaneuve
mons tournai
mons bruxelles
arlon liege
tournai bruxelles
liege gand
gand louvainlaneuve
arlon tournai
```

Output 2

```
impossible
```

Problème D: Difficile Stockage.

Temps limite : 1s

Pour se désaltérer après le concours, l'équipe organisatrice du KARWa a prévu un grand nombre de boissons alcoolisées stockées dans un grand frigo de n emplacements de large et deux emplacements de haut. Vous pouvez stocker chaque boisson de deux façons : soit verticalement, et elle prend ainsi un emplacement de long sur deux emplacements de haut, soit horizontalement, et elle prend ainsi deux emplacements de long sur un emplacement de haut. De combien de manières différentes est-il possible de remplir le frigo ?

Attention, la réponse peut être très grande. Nous vous demandons donc d'output la réponse modulo $7 + 10^9$.

Input :

Un entier n ($1 \leq n \leq 10^{18}$), le nombre d'emplacements dans le frigo sur sa longueur.

Output :

Le nombre de manières de stocker des boissons dans le frigo modulo $7 + 10^9$.

Exemple d'input et son output correspondant :

Input 1	Output 1
`2`	`2`

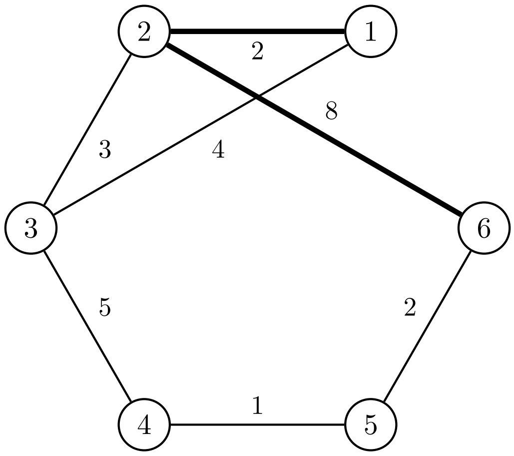
Input 2	Output 2
`4`	`5`

Problème E: En Retard.

Temps limite : 2s

Jérôme et ses deux coéquipiers doivent se rendre aux KARWa à l'UMONS. Malheureusement, ils se sont trompés d'université et ont atterri à l'UCLouvain ! Ils sont pressés par le temps et demandent à leur ami Nicolas, expert des cartes et des plus courts chemins, qui a sa disposition une carte avec n gares et m lignes de trains bidirectionnelles (lignes qui vont dans les deux sens), prenant chacune un temps w. Pouvez-vous déterminer s'ils auront le temps d'arriver à l'heure ?

L'UCLouvain est représentée par l'intersection 1 et l'UMONS par l'intersection n.



Exemple 1 : avec le chemin le plus court de 1 à 6 en gras

Un chemin est garanti entre les deux universités.

Input :

Chaque cas consiste en :

- une ligne de deux entiers n et m ($2 \leq n, m \leq 10^6$), où n est le nombre d'intersections et m le nombre de liaisons,
- m lignes contenant chacune 3 entiers u ($1 \leq u \leq n$), v ($1 \leq v \leq n$), et w ($1 \leq w \leq 10^6$), représentant une liaison bidirectionnelle entre u et v, avec un poids w.

Output :

Une ligne contenant la longueur du plus court chemin entre 1 et n.

Exemple d'input et son output correspondant :

Input

```
6 7
1 2 2
1 3 4
3 4 5
2 6 8
2 3 3
4 5 1
5 6 2
```

Output

```
10
```

Problème F: Facile Stockage.

Temps limite : 1s

Pour se désaltérer après le concours, l'équipe organisatrice du KARWa a prévu un grand nombre de boissons alcoolisées stockées dans un grand frigo de n emplacements de large et deux emplacements de haut. Vous pouvez stocker chaque boisson de deux façons : soit verticalement, et elle prend ainsi un emplacement de long sur deux emplacements de haut, soit horizontalement, et elle prend ainsi deux emplacements de long sur un emplacement de haut. De combien de manières différentes est-il possible de remplir le frigo ?

Attention, la réponse peut être très grande. Nous vous demandons donc d'output la réponse modulo $7 + 10^9$.

Input :

Un entier n ($1 \leq n \leq 10^6$), le nombre d'emplacements dans le frigo sur sa longueur.

Output :

Le nombre de manières de stocker des boissons dans le frigo modulo $7 + 10^9$.

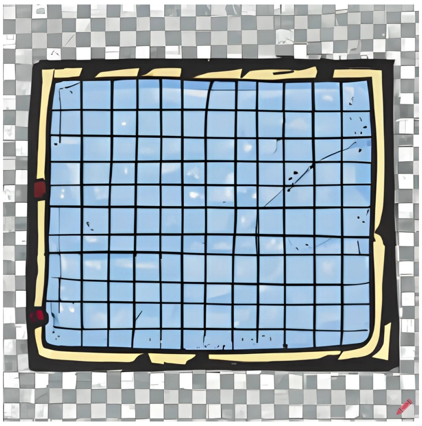
Exemple d'input et son output correspondant :

Input 1	Output 1
2	2

Input 2	Output 2
4	5

Problème G: Ravitaillement.

Temps limite : 5s



Vous êtes fraichement recruté à un poste important de développement et d'optimisation au Centre d'Activités Récréatives des Week-ends Aventureux (CARWA). Le week-end approche à grand pas et une activité a lieu à Delft, les équipes de candidats et de responsables sont déjà parties vers leur destination, mais nous avons un problème, personne n'a pensé au ravitaillement ! La ville de Delft est un véritable dédale, un grand rectangle composée de zones carrées, et il serait dangereux de laisser nos équipes sans un tel support.

Nous avons besoin de votre aide pour trouver établir la taille idéale d'un campement de ravitaillement dans la ville sans perturber le déroulement de l'activité pour les candidats. Vous disposez du nom de chaque équipe, leur position dans la ville et s'il s'agit d'une équipe de candidat ou de responsables. La ville est composées de $n \times m$ petites zones carrées sur votre plan. Une équipe de candidats ne peut se trouver que sur une seule case, mais une seule équipe de responsable peut être éparpillée sur plusieurs cases. Deux équipes ne peuvent jamais se trouver sur la même case mais une case peut être vide.

Votre but est de trouver le plus grand rectangle dans Delft où seules des équipes de responsables se trouvent, afin qu'elles puissent créer le plus grand camps de ravitaillement. Il doit impérativement être de forme rectangulaire, vu la complexité de se déplacer dans Delft. Lorsque vous aurez trouvé ce rectangle, indiquez à vos supérieurs le nombre d'équipes qui seront mobilisées pour établir ce ravitaillement.

Input :

L'entrée consiste en :

- une ligne de trois entiers n, m et k ($1 \leq n, m \leq 10^3$, $1 \leq k \leq 20$), respectivement le nombre de zones en ligne, le nombre de zones en colonne et le nombre d'équipes de responsables présentes sur le terrain,
- n lignes, où une ligne contient m noms séparés par un espace, c'est-à-dire de la forme "`nom1 nom2 ... nom_m`". Le $j^{\text{ème}}$ nom de la $i^{\text{ème}}$ ligne représente l'équipe sur la zone (i, j),
- k lignes, où une ligne est le nom d'une équipe de responsables.

Les noms sont composés de lettres de l'alphabet, en minuscule et sans accent ni caractères spéciaux. Lorsqu'une case n'est occupée par aucune équipe, le nom correspondant sera "`null`".

Output :

Le nombre de zones dans le plus grand rectangle composé uniquement de zones occupées par des responsables.

Exemple d'input et son output correspondant :

Input 1

```
3 5 4
securail null null powerrangers null
null securail null algo rythme
distance waffle delft distance securail
securail
delft
algo
distance
```

Output 1

```
3
```

Input 2

```
1 3 2
useless teams null
useless
teams
```

Output 2

```
2
```

Input 3

```
2 5 2
a a a b b
a a a b a
a
b
```

Output 3

```
10
```

Problème H: Horizon Complexe.

Temps limite : 1s

La grande ville de Karwa-sur-mer souhaite moderniser les buildings présents sur la digue. En effet, la ville veut poser des panneaux solaires sur ceux-ci et, afin de profiter un maximum de la production électrique de ceux-ci, elle décide de recouvrir bien évidemment les toits des bâtiments et les façades de côtés.

Pour cela, vous êtes engagé afin de déterminer combien de mètres de panneaux la ville aura besoin pour équiper ses bâtiments. La ville vous donne pour cela un plan en vue de face de la digue vous spécifiant la taille des bâtiments comme ceci :

Pour chaque emplacement de 1 mètre de large, le plan vous donne la hauteur du bâtiment à cet emplacement comme illustré par la figure ci-dessous.

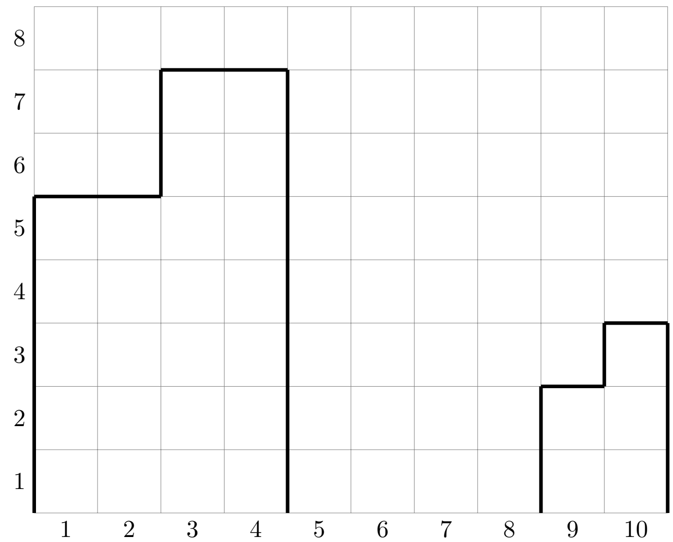


Illustration de l'exemple 1
En noir, là où la ville souhaite poser des panneaux sur les bâtiments.

Input :

Chaque cas consiste en :

- ◇ une ligne avec un entier n ($1 \leq n \leq 10^4$), la longueur en mètres de la digue,
- ◇ une seconde ligne qui représente la ligne d'horizon, composée de n entiers h_i ($0 \leq h_i \leq 100$) séparés d'un espace représentant la hauteur du bâtiment en position i.

Output :

Un entier représentant le nombre de mètres de panneaux nécessaire afin de recouvrir la totalité des côtés et dessus des bâtiments.

Exemple d'input et son output correspondant :

Input 1

```
10
5 5 7 7 0 0 0 0 2 3
```

Output 1

```
26
```

Input 2

```
5
0 2 4 6 8
```

Output 2

```
20
```

Problème I: Impossible.

Temps limite : 1s

Pendant que ses équipes participent au BAPC, Henri, leur coach, s'ennuie. Il décide alors de s'attaquer aux problèmes de la 1ère édition du prestigieux KARWa. L'un des problèmes, "Difficile Stockage" consiste à trouver le nombre de manières que l'on peut disposer des boissons alcoolisées, problème qui fait appel à des concepts avancés d'analyse numérique et d'éléments finis.

Malheureusement, Henri, pourtant si proche de la solution, ne se rappelle plus comment calculer le périmètre d'un carré à partir de son aire. Étant donné l'aire a d'un carré, aidez Henri à calculer son périmètre.

Input :

Une ligne : un entier a ($0 \leq a \leq 123456789$), l'aire du carré.

Output :

Le périmètre d'un carré d'aire a. La réponse doit avoir une erreur absolue ou relative de maximum 10^{-6}.

Exemple d'input et son output correspondant :

Input 1	Output 1
`1`	`4.0`

Input 2	Output 2
`4`	`8.0`

Problème J: Koalation.

Temps limite : 3s

La Koalation Australienne représentée par des Wallabies, également connue sous le nom de KARWa, est une société australienne composée de koalas et de wallabies. Les wallabies étant beaucoup plus intelligents que les koalas, qui ont une capacité limitée en division, ont inventé un système de référencement. Voici comment il fonctionne :

- ◇ Chaque wallaby est représenté par un nombre premier.
- ◇ Chaque koala est représenté par un nombre entier. Il est assigné au wallaby ayant pour numéro le plus petit diviseur de son propre nombre, excepté 1. Par exemple, le koala 15 est assigné au wallaby 3

Cependant, avec le nombre croissant de nouveaux adhérents, les koalas sont devenus difficiles à rassembler selon leur référent. La KARWa a donc besoin de votre aide pour créer un algorithme qui, à partir d'une liste de nombres représentant des koalas, renvoie une nouvelle liste triée en fonction des référents de chaque koala dans l'ordre croissant. Si deux koalas ont le même référent, alors c'est l'ordre croissant qui est utilisé.

Input :

- ◇ un entier n ($1 \leq n \leq 10^4$), le nombre d'éléments dans la liste,
- ◇ une ligne contenant les n nombres n_i, pour i allant de 0 à n ($2 \leq n_i \leq 10^5$).

Output :

Une ligne contenant les n_i triés en fonction de leur plus petit diviseur et dans l'ordre croissant.

Exemple d'input et son output correspondant :

Input 1

```
3
2 5 3
```

Output 1

```
2 3 5
```

Input 2

```
6
6 3 4 5 2 8
```

Output 2

```
2 4 6 8 3 5
```

Problème K: Konnectons Les Tous.

Temps limite : 10s

En prévision de la prochaine édition du BAPC à l'UCLouvain, la société KARWa Corp. souhaite relier tous les karaokés de Belgique entre eux. Pour cela, elle contacte la SNCB et la STIB qui font appel à vous pour trouver la meilleure façon de relier tous les karaokés.

Il y a n karaokés numérotés de 1 à n et m liaisons bidirectionnelles entre certains d'entre eux. Chaque liaison a un poids w représentant la distance entre les deux karaokés reliés. L'objectif est de relier tous les karaokés entre-eux tout en minimisant le coût (distance) total de toutes les liaisons.

Cependant, les petites liaisons (avec un petit poids) sont chères, la SNCB souhaite donc maximiser le poids de l'arête minimale de sorte que le coût (la distance) total des liaisons soit inférieur à 125% du coût (de la distance) minimal pour relier tous les karaokés. Autrement dit, le coût total des liaisons doit être inférieur ou égal à 1.25 fois le coût minimum pour que le projet soit viable.

Il est garanti qu'il est possible de relier tous les karaokés entre eux.

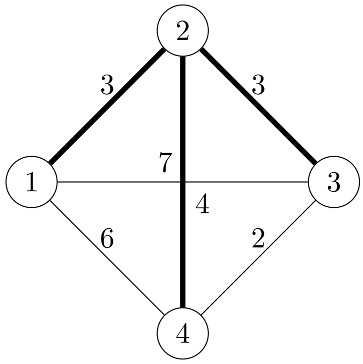


Figure 1: Exemple 1, en gras le réseaux

Input :

L'entrée consiste en :

- une ligne avec 2 entiers n et m ($2 \leq n \leq 10^5$ et $1 \leq m \leq 2 \times 10^5$), représentant respectivement le nombre de karaokés et le nombre de liaisons entre deux karaokés,
- m lignes avec trois entiers u, v et w ($1 \leq u \leq n$, $1 \leq v \leq n$ et $1 \leq w \leq 10^9$), représentant une liaison bidirectionnelle entre deux karaokés u et v dont le poids est w.

Output :

Un entier K représentant le poids de l'arête de poids minimale du réseau connectant les karaokés avec un coût inférieur à 125% du coût de base.

Exemple d'input et son output correspondant :

Input

```
4 6
1 4 6
1 2 3
2 3 3
3 4 2
4 2 4
1 3 7
```

Output

```
3
```

Problème L: Labyrinthe.

Temps limite : 2s

Dimanche matin à 2h30, les équipes de Virgil et d'Aymeric sont en train de rentrer d'un karaoké dans la ville d'Eindhoven. Malheureusement, à cette heure-ci, certaines routes sont bloquées et ils ne savent pas s'ils pourront atteindre l'arrêt de bus. La grille qui représente la ville est de taille $n \times m$, avec n et m compris entre 2 et 1000. Chaque case de la grille peut contenir un "`#`" pour un mur, "`K`" pour les deux équipes, "`B`" pour l'arrêt de bus, et "`.`" pour un emplacement libre. Pouvez-vous déterminer s'il existe un chemin pour atteindre l'arrêt de bus depuis la position des équipes ?

Input :

L'entrée consiste en :

- une ligne avec deux entiers n et m ($2 \leq n, m \leq 2000$), la taille de la grille $n \times m$,
- n lignes de m caractères, représentant la grille.

Output :

"`yes`" s'il existe un chemin permettant d'atteindre l'arrêt de bus depuis les positions des équipes. "`no`" sinon.

Exemple d'input et son output correspondant :

Input 1

```
5 5
#.#.#
#K..#
..#..
##..#
#B..#
```

Output 1

```
yes
```

Input 2

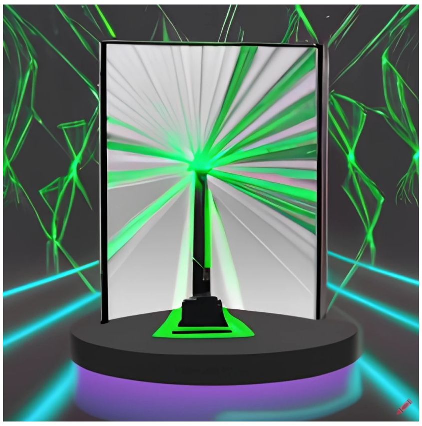
```
5 5
#.#.#
#K..#
..##.
##..#
#B..#
```

Output 2

```
no
```

Problème M: Miroir Magique.

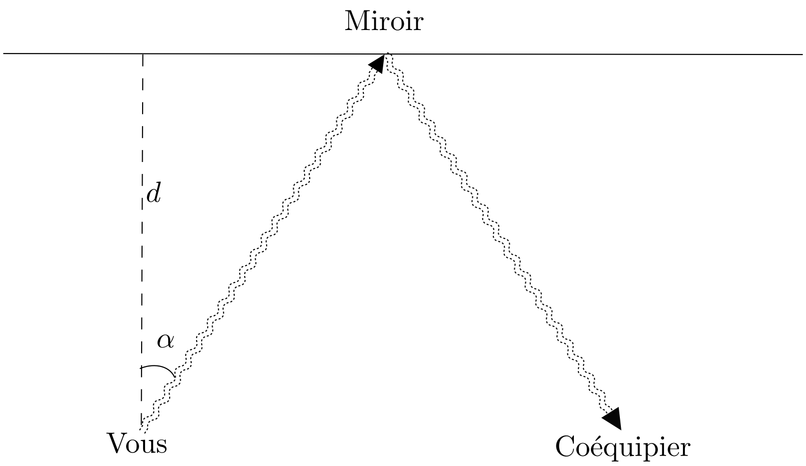
Temps limite : 1s



Ça y est, vous vous préparez pour aller au KARWa-au-quai annuel entre l'UCLouvain et l'UMons, événement incontournable de la vie étudiante, là où les vocations de chanteurs se créent, où tous les labels de musiques viennent repérer les futurs prodiges de demain.

Vous devez être au top de votre forme, tant dans votre chant que pour votre apparence. Rien ne doit être laissé au hasard et vous comptez bien impressionner la totalité des personnes sobres de la salle[1] lorsque vous serez sur scène avec vos coéquipiers pour chanter en coeur. En plus de chanter, vous avez préparé un petit jeu de laser et de lumière via des miroirs pour éblouir la salle.

Sur scène, un de vos coéquipiers et vous serez sur une même ligne, face au public. Derrière vous, un grand miroir sera parallèle à cette ligne. Vous connaissez la distance directe (perpendiculaire) entre votre position et le miroir, et vous allez activer un faisceau laser depuis votre position vers le miroir avec un angle précis par rapport à la perpendiculaire, de sorte que le laser rebondisse sur votre coéquipier. Vous devez déterminer la distance à laquelle votre coéquipier devra se tenir afin qu'il réceptionne parfaitement le faisceau laser.



Input :

L'entrée consiste en :

- une ligne contenant un entier d ($0 < d \leq 10^6$), la distance qui vous sépare du miroir,
- une ligne contenant un entier α ($0 < \alpha < 90$), l'angle en degré de votre laser par rapport à la perpendiculaire vers le miroir.

[1]C'est-à-dire tout le monde, bien évidemment, cet événement est trop important pour boire avant la fin des chansons...

Output :

La distance entre votre coéquipier et vous afin qu'il reçoive le laser sur lui. La réponse doit avoir une erreur absolue ou relative de maximum 10^{-6}.

Exemple d'input et son output correspondant :

Input 1	**Output 1**
`1` `45`	`2.0`

Input 2	**Output 2**
`42` `24`	`37.39920956591704`

Problème N: Navettes.

Temps limite : 0.5s

Alexandre et Virgil organisent une sortie en karaoké pour n participants. Pour se rendre au karaoké depuis le point de départ, ils doivent louer des navettes. Chaque navette peut accueillir m personnes.

Quel est le nombre minimal de navettes que doivent louer Alexandre et Virgil pour transporter tous les participants au karaoké ?

Input :

Deux entiers n ($1 \leq n \leq 10^9$) et m ($1 \leq m \leq 10^6$) où n représente le nombre de participants et m représente la capacité d'une navette.

Output :

L'algorithme doit retourner le nombre minimal de navettes à louer pour transporter tous les participants au karaoké.

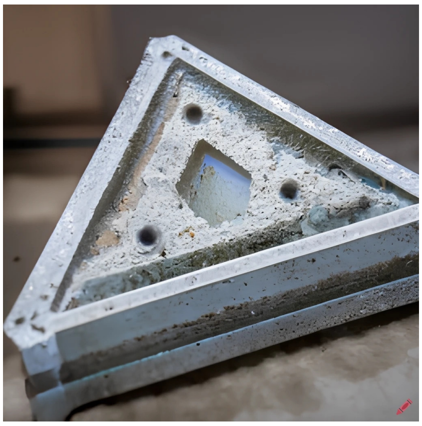
Exemple d'input et son output correspondant :

Input 1	Output 1
`50 9`	`6`

Input 2	Output 2
`420 69`	`7`

Problème O: Palindronium Pythagoricien.

Temps limite : 1s

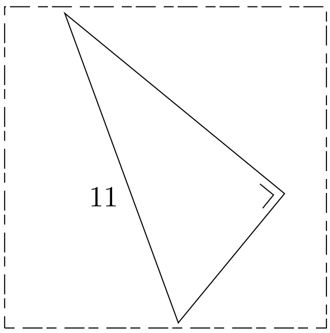


Un débat a éclaté entre les étudiants de l'UMons et de l'UCLouvain. Ils ne parviennent pas à se mettre d'accord sur la meilleure des deux universités, surtout dans le domaine scientifique. Nous avons besoin de votre aide pour les départager et ainsi avantager votre université au sujet d'un tout nouveau métal appelé "palindronium".

Votre mission est d'optimiser la production de palindronium dans votre université, production pour l'instant très coûteuse. Pour l'optimiser, le métal doit être fabriqué dans un moule d'une forme d'un triangle rectangle dont la longueur de l'hypoténuse est un nombre entier et un palindrome. Par souci d'espace, vous ne pouvez modéliser qu'un seul moule, et celui-ci doit pouvoir rentrer dans un carré de taille $n \times n$, il doit ainsi avoir une aire maximale.

Un nombre est un palindrome s'il ne change pas lorsqu'il est lu de gauche à droite ou de droite à gauche, chiffre par chiffre. Par exemple, 191 est un palindrome, mais 155 ne l'est pas. Votre but est d'avoir la plus grande aire possible sous ces contraintes ! Attention cependant, on ne demande qu'à l'hypoténuse d'être entier et d'être un palindrome, les autres côtés n'ont aucune restriction.

Par exemple, si le carré dans lequel votre moule doit rentrer a une taille de 12×12, vous pouvez fabriquer un moule de taille $(3, 4, 5)$ car 5, l'hypothénuse, est un palindrome, ce qui donnera une aire de moule de 6. Vous pouvez aussi faire un moule de taille $(\sqrt{105}, 4, 11)$.



$n = 12$

Input :

Un entier n ($1 \leq n \leq 10^8$) donnant la longueur des côtés du carré devant contenir le moule triangulaire.

Output :

Une ligne contenant l'aire maximale que vous pouvez obtenir avec un moule, à une erreur absolue ou relative de maximum 10^{-6}.

Exemple d'input et son output correspondant :

Input	Output
Input 1	**Output 1**
`15`	`30.25`
Input 2	**Output 2**
`100`	`4970.25`