

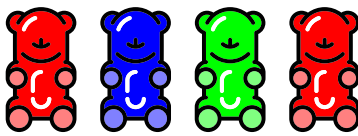


Gummibären

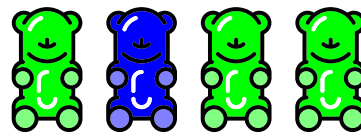
Maus Stofl mag Gummibären, und er mag es, sie in einer besonderen Art zu essen. Ein Pack Gummibären enthält n Gummibären. Jeder Gummibär hat eine Geschmacksrichtung, die als Ganzzahl gegeben ist.

Stofl will das Pack essen, indem er wiederholt Happen von k Gummibären isst (es ist garantiert, dass k ein Teiler von n ist). Ein Happen Gummibären schmeckt *lecker*, wenn sich eine dominante Geschmacksrichtung durchsetzt, d.h. es gibt eine Geschmacksrichtung, sodass strikt mehr als $k/2$ Gummibärchen dieses Geschmacks im Happen enthalten sind. Setzt sich keine dominante Geschmacksrichtung durch, schmeckt der Happen *eklig*.

Beispielsweise muss für $k = 4$ ein Happen strikt mehr als $k/2 = 2$, also mindestens 3 Gummibärchen derselben Geschmacksrichtung enthalten.



Igitt! Es gibt keine dominante Geschmacksrichtung.



Mjam! Grün ist die dominante Geschmacksrichtung.

Kann Stofl alle Gummibärchen essen, sodass alle Happen lecker sind? Falls ja, gib eine mögliche Art an, sie zu essen.

Eingabe

Die erste Zeile der Eingabe enthält drei Ganzzahlen n , k und p – die Anzahl Gummibärchen n , die Grösse eines Happens k und p , das angibt, ob du auch ausgeben musst, wie du die Gummibärchen isst (wenn $p = 1$, dann musst du das ausgeben, wenn $p = 0$, dann nicht).

Die zweite Zeile enthält n Ganzzahlen f_i – die Geschmacksrichtungen der Gummibären.

Ausgabe

Falls es möglich ist, die Gummibären so zu essen, dass alle Happen lecker sind, gib auf der ersten Zeile "Yummy!" aus. Andernfalls gib "Yuck!" aus.

Falls $p = 1$ und deine erste Zeile war "Yummy!", solltest du auch eine mögliche Art ausgeben, die Gummibären zu essen. Gib dazu n/k Zeilen aus, die je k Zahlen enthalten: Die i -te Zeile sollte die Geschmacksrichtungen der Gummibären enthalten, die im i -ten Happen enthalten sein soll. Du kannst jede Geschmacksrichtung so oft verwenden, wie sie in der Eingabe vorkommt.

Bei $p = 0$ werden alle Zeilen bis auf die erste ignoriert.

Wenn mehrere Lösungen existieren, kannst du eine beliebige ausgeben.

Limits

Es gibt acht Testgruppen. In allen Testgruppen gilt $1 \leq k \leq n$.

In ungeraden Testgruppen (1, 3, 5 und 7) gilt $p = 0$.

In geraden Testgruppen (2, 4, 6 und 8) gilt $p = 1$.

In den Beispielen gilt $p = 1$. Falls du nur Testgruppen mit $p = 0$ lösen willst, kannst du in der Bewertung trotzdem sehen, ob die erste Zeile korrekt ist.

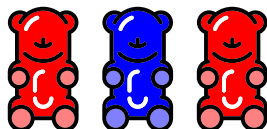
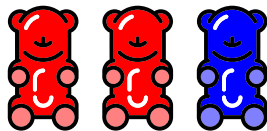
- Testgruppen 1 und 2, wert je 10 Punkte: $n \leq 100\,000$, $0 \leq f_i \leq n/k$ und jede Geschmacksrichtung kommt genau k mal vor. In anderen Worten enthalten die Werte f_i genau k mal 0, k mal 1, etc., und k mal $n/k - 1$.

- Testgruppen 3 und 4, wert je 10 Punkte: $n \leq 100\,000$ und $0 \leq f_i < 2$, d.h. es gibt höchstens 2 unterschiedliche Geschmacksrichtungen.
- Testgruppen 5 und 6, wert je 10 Punkte: $n \leq 1000$ und $0 \leq f_i < n$
- Testgruppen 7 und 8, wert je 20 Punkte: $n \leq 100\,000$ und $0 \leq f_i < 10^9$

Beispiele

| Eingabe | Ausgabe |
|----------------------|--------------------------|
| 6 3 1 1 0 0 0 0 1 | Yummy! 0 0 1 0 1 0 |

Diese Eingabe wäre für Teilaufgaben 3, 4, 5, 6, 7 und 8 gültig, wenn wir den Wert von p nicht berücksichtigen. Die Lösung sieht wie folgt aus (mit rot = 0 und blau = 1):



| Eingabe | Ausgabe |
|---|--|
| 16 4 1 0 0 0 1 1 1 1 1 1 1 1 3 3 4 4 | Yummy! 0 0 0 3 1 1 1 3 1 1 1 4 1 1 1 4 |

Diese Eingabe wäre für Teilaufgaben 5, 6, 7 und 8 gültig, ohne Rücksicht auf p .

| Eingabe | Ausgabe |
|-----------------------------------|---------|
| 12 3 1 0 0 0 0 0 1 2 2 2 3 4 5 | Yuck! |

Diese Eingabe wäre für Teilaufgaben 5, 6, 7 und 8 gültig, ohne Rücksicht auf p .

| Eingabe | Ausgabe |
|------------------|---------|
| 4 2 1 0 1 0 0 | Yuck! |

Diese Eingabe wäre für Teilaufgaben 3, 4, 5, 6, 7 und 8 gültig, ohne Rücksicht auf p .

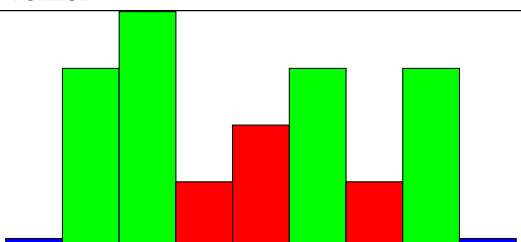
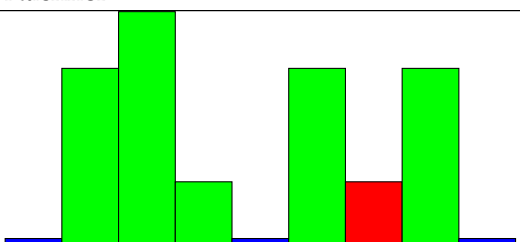
| Eingabe | Ausgabe |
|-----------------------------------|---|
| 12 4 1 0 1 2 2 1 1 1 0 2 2 0 0 | Yummy! 0 0 0 0 1 1 1 1 2 2 2 2 |

Diese Eingabe wäre für Teilaufgaben 1, 2, 5, 6, 7 und 8 gültig, ohne Rücksicht auf p .

Wassersicht

Maus Stofl baut einen Wanderweg zwischen zwei Seen. Der Weg besteht aus n Abschnitten, wobei Abschnitt i Höhe h_i hat. Die zwei Seen sind auf Höhe 0. Sie befinden sich vor dem 0-ten bzw. nach dem $(n - 1)$ -ten Abschnitt. Stofl möchte den Wanderweg möglichst schön machen. Eine Abschnitt i des Wanderwegs ist *schön*, wenn man einen See sehen kann, d.h. wenn es einen See gibt, sodass kein Abschnitt zwischen i und diesem See höher als h_i ist. Die Schönheit des Wanderweges ist gleich der Anzahl schöner Abschnitte, die er enthält. Stofl kann genau einen Abschnitt durch einen weiteren See ersetzen, der ebenfalls auf Höhe 0 liegt. Dieser See zählt auch als schöner Abschnitt. (Da der Wanderweg keine Lücken haben darf, baut Stofl eine Brücke über den neuen See, von der man natürlich den See sieht.)

Unten ist gezeigt, wie es aussehen könnte. Die schönen Abschnitte sind grün markiert.

| Vorher | Nachher |
|---|--|
|  <p>Dies ist die Situation bevor der neue See gebaut wurde. Drei der Abschnitte haben keine Sicht auf einen See, deshalb ist die Schönheit des Wanderweges 4.</p> |  <p>Nachdem ein neuer See in der Mitte gebaut wurde, sind alle Abschnitte bis auf eines schön geworden und die Schönheit des Wanderweges ist nun 6.</p> |

Eingabe

Die erste Zeile der Eingabe enthält eine Ganzzahl n , die Anzahl Abschnitte. Die nächste Zeile enthält n Ganzzahlen h_i , die Höhen der Abschnitte.

Ausgabe

Gib eine einzelne Ganzzahl aus: die maximale Anzahl schöner Abschnitte, die Stofl erreichen kann.

Limits

In allen Testgruppen gilt $1 \leq h_i \leq 10^9$ und $1 \leq n \leq 500\,000$.

- In der ersten Testgruppe, wert 30 Punkte, $1 \leq n \leq 500$.
- In der zweiten Testgruppe, wert 30 Punkte, $1 \leq n \leq 10\,000$.
- In der dritten Testgruppe, wert 40 Punkte, gibt es keine weiteren Einschränkungen.

Beispiele

| Eingabe | Ausgabe |
|------------|---------|
| 3 2 1 3 | 3 |

Stofl kann einen beliebigen der drei Abschnitte durch einen See ersetzen, damit alle Abschnitte schön sind.



| Eingabe | Ausgabe |
|--------------------|----------------|
| 7 3 4 1 2 1 3 2 | 7 |

Der Abschnitt mit Höhe 2 ist der einzige Abschnitt, den Stofl durch einen See ersetzen kann, damit alle Abschnitte schön sind.

| Eingabe | Ausgabe |
|--------------------|----------------|
| 7 3 4 1 2 3 1 3 | 6 |

Stofl kann nicht alle Abschnitte schön machen. Wenn er das Segment mit Höhe 2 mit einem See ersetzt, sind sechs Abschnitte schön. Dieses Beispiel ist im Bild oben dargestellt.

Rutschende Maus

Maus Daniel steht in der Mitte eines Eisfeldes. Das Eisfeld ist ein rechteckiges Gitter der Grösse $n \times m$, welches aus drei Arten von Zellen besteht:

- Wand "#": Maus Daniel kann sich nicht auf diese Zelle bewegen.
- Eis "+": Wenn er sich auf solch eine Zelle bewegt, rutscht Maus Daniel darüber, sodass er die Zelle in die gleiche Richtung verlässt (ausser es ist eine Wand in dieser Richtung, dann stoppt er).
- Erde ".": Wenn Maus Daniel sich auf solch eine Zelle bewegt, stoppt er. Er kann die Zelle in jede der 4 Richtungen verlassen.

Es gibt zwei spezielle Erdzellen:

- Startzelle "d": Maus Daniel beginnt hier.
- Zielzelle "g": Maus Daniel gewinnt, wenn er diese Zelle erreicht.

Maus Daniel bewegt sich mit einer Geschwindigkeit von einer Zelle pro Sekunde. Kannst du ihm helfen, das Ziel möglichst schnell zu erreichen?

Eingabe

Die erste Zeile enthält zwei ganze Zahlen n und m . Danach folgen m Zeilen mit je n Zeichen. Jedes Zeichen ist entweder #, +, ., d oder g. d und g gibt es genau einmal. Alle Randzellen sind #.

Ausgabe

Gib aus, wie viele Sekunden Maus Daniel braucht, um das Ziel zu erreichen, oder "IMPOSSIBLE", falls das Ziel unerreichbar ist.

Limits

Es gibt 4 Teilaufgaben, die je 25 Punkte geben.

- In Teilaufgabe 1 gilt $n, m \leq 100$ und es gibt keine Eiszellen.
- In Teilaufgabe 2 gilt $n, m \leq 100$.
- In Teilaufgabe 3 gilt $n, m \leq 1000$.
- In Teilaufgabe 4 gilt $n, m \leq 2000$.

Beispiele

| Eingabe | Ausgabe |
|--|----------------|
| <pre> 8 5 ##### #.++g#.# #.+++ .d# #..+..+# ##### </pre> | <pre> 5 </pre> |

Daniel kann nicht nach oben und dann links gehen, weil eine Wand im Weg ist. Wenn er zwei Schritte nach links geht, würde er bis an die linke Wand übers Eis rutschen, und er würde 9 Sekunden brauchen um das Ziel zu erreichen. Ein schnellerer Weg ist, zuerst nach unten zu laufen (er wird von der Wand gebremst), dann zwei Schritte nach links und dann übers Eis bis ins Ziel rutschen.



| Eingabe | Ausgabe |
|--------------------------------|----------------|
| 5 3 ##### #d#g# ##### | IMPOSSIBLE |

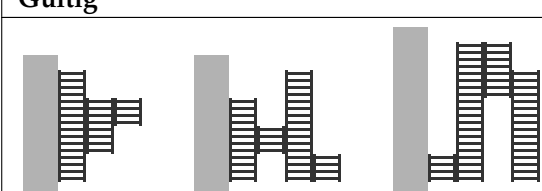
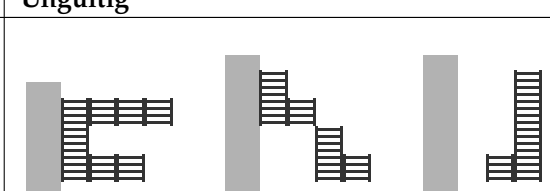
Leider ist das Ziel hinter einer Wand, sodass Maus Daniel es nicht erreichen kann.

Leiterbalkon

Maus Binna hat gerade einen riesigen Turm fertig gebaut, und nun will sie noch einen Balkon anbauen. Sie will nicht einfach einen gewöhnlichen Balkon, nein, der Balkon soll aus Leitern gemacht sein. Das wird es ihr ermöglichen, nach Belieben rauf und runter zu klettern und dabei den Sonnenuntergang zu geniessen.

Wenn wir uns den Turm auf der linken Seite vorstellen und den Sonnenuntergang rechts in der Mitte, dann können die Leitern folgendermassen installiert werden: Die Leiter ganz links muss am Turm angeschraubt sein, und jede weitere Leiter muss an die Leiter direkt links von ihr befestigt werden. Um zwei Leitern aneinander zu schrauben, müssen sich die Seiten berühren, es reicht nicht wenn die Ecken aufeinandertreffen (sonst hält das ganze nicht).

Hinzu kommt, dass nie zwei Leitern übereinander montiert werden dürfen mit einer Lücke dazwischen, weil Maus Binna fallen und sich ein Bein brechen könnte, wenn sie die Lücke übersieht.

| Gültig | Ungültig |
|--|---|
|  |  |

Der Turm ist H Mausmeter hoch, und Maus Binna schätzt, dass der Turm standhält, wenn die Leitern bis zu W Mausmeter nach rechts in die Luft ragen. Sie hat bereits eine Leiter gekauft, welche N Mausmeter lang ist und 1 Mausmeter breit. Sie kann die Leiter in verschiedene Stücke zersägen, aber aus ästhetischen Gründen muss die Länge jedes Stückes eine ganze Zahl Mausmeter sein. Die Stücke wird sie dann zusammenschrauben, um ihren Balkon zu bauen.

Mit einer Drohne hat Maus Binna bereits ausgemessen, wie viel Sonnenlicht auf die $H \times W$ Fläche vor dem Turm trifft. Sie will Summe des Sonnenlichts maximieren, die der Balkon erhält. Kannst du ihr helfen, dieses Ziel zu erreichen?

Eingabe

Die erste Zeile enthält drei ganze Zahlen H , W und N – die Höhe des Turms, die maximale Distanz, die der Balkon nach aussen ragen darf, und die Länge der Leiter in Mausmeter, die Binna zur Verfügung steht. Die nächsten H Zeilen enthalten je W Ganzzahlen $a_{i,j}$, die Sonnenlicht-Messungen vor dem Turm.

Ausgabe

Gib eine einzige Ganzzahl aus – die maximale Summe des Sonnenlichtes, die Maus Binna auf ihrem Balkon haben kann.

Limits

In jedem Testfall gilt $1 \leq H, W, N$ und $N \leq H \cdot W$ und $0 \leq a_{i,j} \leq 10^6$.

- In der ersten Teilaufgabe (20 Punkte) gilt $H, W, N \leq 20$.
- In der zweiten Teilaufgabe (20 Punkte) gilt $H, W, N \leq 40$.
- In der dritten Teilaufgabe (20 Punkte) gilt $H, W, N \leq 80$.
- In der vierten Teilaufgabe (20 Punkte) gilt $H \leq 100, W \leq 80$ und $N \leq 100$.



- In der fünften Teilaufgabe (10 Punkte) gilt $H \leq 120$, $W \leq 50$ und $N \leq 120$.
- In der sechsten Teilaufgabe (10 Punkte) gilt $H \leq 130$, $W \leq 30$ und $N \leq 130$.

Beispiele

| Eingabe | Ausgabe |
|-------------------------|---------|
| 2 3 3 1 2 3 7 8 9 | 24 |

| Eingabe | Ausgabe |
|---|---------|
| 7 6 10 2 6 1 1 2 9 6 2 9 1 9 3 7 5 6 8 2 6 5 6 5 4 5 4 6 7 3 6 3 8 4 1 4 5 2 1 3 2 5 4 6 3 | 65 |