

public static void main(Strin

>>[>>+>+<<<-]>[>[<<+>>-]<->>[<<<<+>>>+>-]<<<  
<[>>>>+<<<<-]>>>>>

<p><b>Code Golf</b></p> <p><b>Projektwoche 2010</b></p> <p>2010-05-25</p> <p>Johannes Rössel, Hannes Müller</p>
---

for x in"\*  
i=".NI5VF  
s+=b[i:i+9]+"w" ({:c)&+@:++:@:({.\*{:})

<b>Zeilennummern</b>	<b>2</b>
<b>Würfel</b>	<b>3</b>
<b>All your base are belong to us</b>	<b>4</b>
<b>Verlauf</b>	<b>5</b>
<b>Werkzeuge</b>	<b>7</b>
<b>Blöcke</b>	<b>8</b>
<b>Vigenère</b>	<b>11</b>
<b>Pfade</b>	<b>13</b>

var l=s.Split('\n');n\*=2;for(int i=0,j,c;i<1  
\$\_=()=>A..\$\_ 26#.64-~a.i.  
[0]\+{31&\26\*+}\* =:>((<:;|::0\$])((2\$>:@+:  
)\$-%~i:)300

{:99&<.@;@(({:+4&<@:({.+1&{)@:\*:);~(0{c)&+@(  
r 10 r \ @ . \* . ( 1 r ~ \ @ . . . @ / r \* 10 r ) \ ^ . 00 c

## 1. Zeilennummern

Die Eingabe soll mit Zeilennummern versehen werden.

### Eingabe

Beliebiger Text in potentiell mehreren Zeilen.

### Ausgabe

Der gleiche Text, nur mit Zeilennummern vor jeder Zeile. Die Nummern fangen mit 1 an und werden mit einem Leerzeichen vom Rest der Zeile getrennt.

### Beispieleingabe

```
Das  
ist  
ein  
Text  
in  
mehreren  
Zeilen  
der  
mit  
Zeilennummern  
versehen wird
```

### Beispielausgabe

```
1 Das  
2 ist  
3 ein  
4 Text  
5 in  
6 mehreren  
7 Zeilen  
8 der  
9 mit  
10 Zeilennummern  
11 versehen wird
```

## 2. Würfel

Es soll ein Würfel (Kubus, nicht Spielwürfel) in ASCII-Art gezeichnet werden. Als Perspektive ist eine ungefähre Kavaliersperspektive vorgesehen.

Da nicht-proportionale Schriften meist etwa doppelt so hohe wie breite Zeichen haben, gibt die Eingabe die Länge der vertikalen Seitenkanten (ohne Ecken) an. Horizontale Kanten werden deswegen doppelt so lang gezeichnet (damit als Ergebnis ungefähr ein Würfel entsteht). Die schrägen Kanten nach hinten werden bei der Kavaliersperspektive um die Hälfte verkürzt gezeichnet. Jede Ecke des Würfels wird durch ein + markiert, horizontale Kanten durch - dargestellt, vertikale Kanten durch | und schräge Kanten durch /.

Zusammenfassend: Bei einer Eingabe von  $n$  gilt folglich:

- Horizontale Kanten verwenden ein Bindestrich-Minus (-) und bestehen aus  $2n$  Zeichen.
- Vertikale Kanten verwenden einen Senkrechtstrich (|) und bestehen aus  $n$  Zeichen.
- Schräge Kanten verwenden einen Vorwärts-Schrägstrich (/) und bestehen aus  $\frac{n}{2}$  Zeichen.
- Ecken des Würfels verwenden das Plus-Zeichen (+). Ecken zählen nicht zur Länge einer Kante, also besteht eine horizontale Kante für  $n = 2$  aus  $2n + 2 = 6$  Zeichen (siehe Beispiele).

### Eingabe

Die einzige Eingabe ist eine positive, gerade Zahl  $n$  ( $2 \leq n \leq 30$ ), die die Seitenlänge des Würfels angibt, gefolgt von einem Zeilenumbruch.

### Ausgabe

Die Ausgabe stellt einen Würfel nach oben dargelegten Regen dar.

### Beispieleingabe

| 2

| 4

### Beispielausgabe

```
|  +---+
|  /   /
|  +---+
|  |   |
|  |   |
|  +---+
```

```
|  +---+
|  /   /
|  +---+
|  |   |
|  |   |
|  +---+
```

### 3. All your base are belong to us

Gegeben ist eine Reihe von Ziffern. Hierbei ist jede Ziffer (bis auf die erste) zu interpretieren als die Zahlensystembasis für die jeweils vorhergehende. Jede Zahl wird in die entsprechende Basis konvertiert und das Ergebnis als Dezimalzahl interpretiert. Gesucht ist die Summe der zurückkonvertierten Dezimalzahlen.

Die gegebene Ziffernfolge  $a_1, a_2, a_3, \dots, a_n$  wird von links nach rechts betrachtet; jedes  $a_i$  ( $i < n$ ) wird in die Basis  $a_{i+1}$  umgerechnet. Die resultierende Zahl wird als Dezimalzahl interpretiert und von allen die Summe gebildet.

Beispielhaft sei folgende Eingabe gegeben:

| 679342

Wir erhalten also die Ziffernfolge (6, 7, 9, 3, 4, 2). Jede Ziffer außer der ersten wird nun als Basis der vorhergehenden interpretiert. Wir erhalten also folgende Zahlenfolge:

$$([6]_7, [7]_9, [9]_3, [3]_4, [4]_2)$$

Und nach Umrechnung in die jeweiligen Basen folgende Zahlen:

$$(6, 7, 100, 3, 100)$$

Die Summe selbiger ergibt dann die erwartete Ausgabe:

| 216

#### Eingabe

Gegeben wird eine Folge von Ziffern:  $a_1, a_2, a_3, \dots, a_n$  ( $n > 1$ ), wobei jedes  $a_i \in \{2, 3, \dots, 9\}$ . Damit sind (für unsere Zwecke) unsinnige Basen wie 0 oder 1 ausgeschlossen sowie solche, die zu Nicht-Dezimalziffern führen. Die Eingabe besteht nur aus den genannten Ziffern, ohne Trennzeichen wie Leerzeichen, o. ä. Nach den Ziffern folgt ein einzelner Zeilenumbruch.

#### Ausgabe

Ausgabe ist eine einzelne Zahl (zur Basis 10), die die Summe der Dezimalinterpretationen der jeweiligen Basenumrechnungen darstellt.

#### Beispieleingabe

| 679342

| 23456789

| 98765432

| 98346573494

| 29292929292

#### Beispielausgabe

| 216

| 35

| 77

| 105

| 5015

## 4. Verlauf

Es soll ein kreisförmiger »Zeichenverlauf« (wie ein Farbverlauf, beginnend in einem Mittelpunkt und sich nach außen hin fortsetzend) in ASCII gezeichnet werden. Hierbei ist die Größe des Ausgaberas- ters vorgegeben, der Mittelpunkt, sowie die für den Verlauf zu verwendenden Zeichen werden je- doch eingelesen.

Der Verlauf wird in einem Raster von 70×25 Zeichen ausgegeben. Jedes Zeichen in diesem Raster wird folgendermaßen ausgewählt:

- Die obere linke Ecke des Rasters hat die Koordinaten (0, 0).
- Da proportionale Schriften meist etwa doppelt so hohe wie breite Zeichen haben, muss die Abstandsberechnung eines Zeichens vom Mittelpunkt des Verlaufes die *horizontale* Distanz halbieren. Dies stellt sicher, dass das Ergebnis ungefähr kreisförmig ist.
- Der Endpunkt des Verlaufes (also der Punkt, ab dem nur noch das letzte Zeichen des Verlau- fes auftritt) befindet sich in einem Abstand von 35 vom Mittelpunkt des Verlaufes.
- Der Abstand wird daher auf die Länge der zu verwendenden Liste von Zeichen skaliert und entsprechend beschränkt. Es wird abgerundet.

## Eingabe

Die Eingabe sind drei Zeilen:

- Die  $x$ -Koordinate des Verlaufsmittelpunktes.
- Die  $y$ -Koordinate des Verlaufsmittelpunktes.
- Die zu verwendenden Zeichen. Dies ist eine einzelne Zeichenfolge, die ohne Trennzeichen die zu verwendenden Zeichen enthält. Die maximale Länge sind 100 Zeichen.

## Ausgabe

Die Ausgabe ist der gewünschte Verlauf nach oben definierten Regeln. Jede Zeile des Verlaufes wird mit einem Zeilenumbruch abgeschlossen und sonst dürfen keine anderen Zeichen als die in der Ein- gabe geforderten auftauchen.



## 5. Werktage

Gesucht wird der prozentuale Anteil von Werktagen in einem bestimmten Monat. Als Werktage zählen Montag bis Freitag, ohne Beachtung eventueller Feiertage. Es wird der gregorianische Kalender verwendet.

### Eingabe

Die Eingabe ist ein Jahr und Monat gemäß ISO 8601, also `YYYY-MM`. Das Jahr ist immer vierstellig, der Monat immer zweistellig. Das Jahr liegt nicht vor 1582.

### Ausgabe

Die Ausgabe ist der prozentuale Anteil der Werktage (nach obiger Definition) im entsprechenden Monat als ganze Zahl gerundet. Es folgt kein Prozentzeichen oder eventuelle Nachkommastellen.

### Beispieleingabe

2010-05	2010-06	1920-10	2817-12
---------	---------	---------	---------

### Beispielausgabe

68	73	68	68
----	----	----	----

## 6. Blöcke

Gegeben ist ein  $20 \times 20$ -Raster nicht-negativer ganzer Zahlen. Gesucht ist ein Teilbereich der Größe  $3 \times 3$ , für den das Produkt der Summe der Werte in den drei Zeilen maximal wird. Anschaulich dargestellt bedeutet das für den folgenden Teilbereich:

$$\mathfrak{A} = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix}$$

das Ergebnis

$$f(\mathfrak{A}) = \begin{pmatrix} \overbrace{(x_{11} + x_{12} + x_{13})} \\ \cdot \\ \overbrace{(x_{21} + x_{22} + x_{23})} \\ \cdot \\ \overbrace{(x_{31} + x_{32} + x_{33})} \end{pmatrix} \\ = (x_{11} + x_{12} + x_{13}) \cdot (x_{21} + x_{22} + x_{23}) \cdot (x_{31} + x_{32} + x_{33})$$

Die einzelnen Zeilen werden summiert, diese Einzelsummen multipliziert.

An einem Beispiel: Gegeben die folgende Eingabe (hier nur  $5 \times 5$ ):

40	30	42	222	74
294	97	35	272	167
52	8	163	270	242
247	130	216	68	266
283	245	164	53	148

so stellt der rot hervorgehobene Teil die Lösung dar, da hier das Ergebnis maximal wird:

$$(35 + 272 + 167) \cdot (163 + 270 + 242) \cdot (216 + 68 + 266) = 175972500$$

### Eingabe

Die Eingabe sind 20 Zeilen mit je 20 nicht-negativen ganzen Zahlen, jeweils durch ein Leerzeichen getrennt, die oben erwähnten Zahlenraster darstellen. Die Zahlen sind klein genug, dass ein vorzeichenbehafteter 32-bit-Ganzzahltyp ausreicht, um das Ergebnis zu berechnen.

### Ausgabe

Die Ausgabe ist das oben definierte Ergebnis  $f(\mathfrak{A})$  für den  $3 \times 3$ -Teilbereich  $\mathfrak{A}$ , bei dem das Ergebnis maximal wird.



## Beispieleingabe

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

## Beispielausgabe

| 3375

## Beispieleingabe

```
40 30 42 222 74 265 93 209 115 274 139 177 7 274 12 15 103 36 236 91
294 97 35 272 167 126 18 262 292 48 8 296 85 93 245 4 4 240 276 153
52 8 163 270 242 224 142 72 99 240 199 26 224 198 96 242 295 70 56 247
247 130 216 68 266 142 93 214 30 8 12 163 59 84 40 287 233 65 30 242
283 245 164 53 148 282 73 186 296 3 9 233 184 30 205 221 92 96 5 101
132 228 43 91 228 37 266 140 159 109 10 230 40 114 264 3 266 164 219 283
70 207 218 28 299 78 279 30 179 118 196 138 61 229 110 55 203 73 124 112
16 232 28 187 292 78 194 70 65 203 255 227 176 21 32 225 11 15 92 151
58 237 261 41 213 171 170 111 4 209 99 194 40 108 267 137 179 31 35 221
184 209 264 275 163 268 261 40 198 185 45 188 280 273 54 79 270 286 273 121
208 83 66 156 104 62 188 68 53 194 46 279 280 170 266 148 272 285 178 245
210 130 213 118 165 210 213 66 54 189 166 193 57 213 14 101 143 109 172 101
80 193 287 4 140 65 208 111 8 206 107 285 109 29 211 78 170 247 290 193
148 123 15 164 28 153 222 67 156 165 6 163 114 77 165 17 143 209 278 100
3 102 58 148 82 181 84 29 2 236 231 195 118 278 252 257 179 123 276 287
143 141 254 142 200 243 171 32 164 195 235 260 269 191 190 46 65 166 82 146
69 194 65 220 234 110 45 135 125 208 138 20 233 291 256 162 148 216 247 138
10 53 164 107 2 270 226 227 88 206 193 13 41 130 218 249 76 35 207 91
199 36 207 256 58 215 28 277 234 29 198 148 219 244 136 16 30 258 219 264
183 118 48 218 15 125 279 103 73 8 86 113 9 157 239 273 146 208 50 86
```

## Beispielausgabe

| 328536000

## Beispieleingabe

```
242 248 292 60 9 45 94 97 294 202 219 118 115 243 207 81 288 289 241 232
236 225 205 227 242 85 139 63 5 162 170 121 208 133 22 235 125 288 209 270
31 182 108 170 88 268 297 66 249 158 285 157 267 117 150 18 44 66 117 223
205 32 93 132 33 99 209 282 222 272 255 1 80 270 202 54 117 35 139 273
87 74 113 146 177 14 154 92 282 4 192 60 171 286 66 299 89 276 138 289
60 16 143 277 202 284 77 296 215 96 200 10 226 143 136 131 218 246 254 20
244 118 299 218 81 195 129 93 205 202 264 141 196 150 214 72 231 136 243 192
236 157 176 187 104 182 134 29 151 234 81 143 22 119 45 241 17 225 197 7
156 284 92 13 131 60 113 77 228 65 200 83 3 63 83 88 241 113 115 198
62 101 242 270 121 122 119 78 105 273 55 7 239 236 37 252 66 164 56 44
70 57 100 87 34 298 140 259 36 257 1 204 110 299 245 56 43 121 192 10
240 36 272 255 10 194 143 66 27 131 22 78 57 71 128 5 1 155 236 122
281 160 42 147 272 151 196 240 291 280 209 245 271 46 103 35 85 145 78 140
155 74 232 205 235 223 147 39 171 240 56 187 184 70 28 81 293 125 283 159
297 203 75 46 221 77 106 12 268 94 220 156 78 97 266 208 228 137 212 49
4 157 69 51 225 23 61 202 19 23 41 260 161 218 142 251 299 187 283 158
118 136 71 77 21 199 110 87 103 120 153 157 213 234 155 141 135 24 120 199
16 204 292 245 54 260 294 159 254 15 209 41 154 54 231 167 87 291 31 26
212 274 99 199 170 184 47 227 64 2 117 275 67 84 143 214 143 125 24 61
205 250 133 88 210 112 4 160 3 287 54 143 293 94 287 42 105 94 76 169
```

## Beispielausgabe

```
| 311042813
```

## 7. Vigenère

Gegeben sind ein Schlüssel, sowie ein verschlüsselter Text. Mit Hilfe des Schlüssels soll der Text entschlüsselt werden. Als Verschlüsselung kommt die sogenannte Vigenère-Verschlüsselung zum Einsatz.

Hierbei wird der gegebene Schlüssel zunächst so oft wiederholt bis er die Länge des Textes hat. Ist der Klartext also 8 Zeichen lang und lautet der Schlüssel KEY, so muss der Schlüssel auf KEYKEYKE erweitert werden. Danach stellt jedes Zeichen des so verlängerten Schlüssels eine Verschiebung der Zeichen des ursprünglichen Klartextes dar. Wie weit ein Zeichen verschoben wird, hängt von seiner Position im Alphabet ab. Für diese Aufgabe sei das folgende Alphabet zu verwenden:

| abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

Hilfsmittel für die Ver- sowie Entschlüsselung ist das sogenannte Vigenère-Quadrat. Hier wird das Alphabet in zwei Achsen aufgetragen und der Bereich dazwischen durch Alphabet ersetzt, welches für jede Zeile um ein Zeichen weiter nach links verschoben wird. hier vereinfacht für ein kleineres Alphabet (ABCDEFGH) dargestellt:

		Text							
		A	B	C	D	E	F	G	H
Schlüssel	A	A	B	C	D	E	F	G	H
	B	B	C	D	E	F	G	H	A
	C	C	D	E	F	G	H	A	B
	D	D	E	F	G	H	A	B	C
	E	E	F	G	H	A	B	C	D
	F	F	G	H	A	B	C	D	E
	G	G	H	A	B	C	D	E	F
	H	H	A	B	C	D	E	F	G

Verschlüsselter Text

Man hat nun einen Text, der verschlüsselt werden soll, sowie einen Schlüssel der nach obiger Wiederholung genauso lang wie der Text ist. Um den Text nun zu verschlüsseln wird die Zeile ermittelt, die dem jeweils nächsten Zeichen des Schlüssels entspricht sowie die Spalte, die dem entsprechenden Zeichen des Klartextes entspricht. Der entsprechende Buchstabe des verschlüsselten Textes lässt sich nun am Kreuzungspunkt dieser Zeile und Spalte ablesen. Mit obigem Beispielalphabet, dem Schlüssel CAFE und dem Klartext AFFE ergibt sich also der verschlüsselte Text CFCA.

Näheres auch auf Wikipedia<sup>1</sup>, allerdings ist zu beachten, dass hier ein anderes Alphabet verwendet wird.

<sup>1</sup> [http://de.wikipedia.org/wiki/Polyalphabetische\\_Substitution#Vigen.C3.A8re-Verschl.C3.BCsselung](http://de.wikipedia.org/wiki/Polyalphabetische_Substitution#Vigen.C3.A8re-Verschl.C3.BCsselung)

## Eingabe

Zwei Zeilen, wovon die erste den Schlüssel darstellt und die zweite den nach obiger Anleitung verschlüsselten Text.

## Ausgabe

Eine Zeile, die den entschlüsselten Text enthält.

### Beispieleingabe

```
| Key  
| miQ2eE0
```

### Beispielausgabe

```
| Message
```

### Beispieleingabe

```
| GenausolangerKey  
| amALuwBreEkvaOBR
```

### Beispielausgabe

```
| EinLaengererText
```

### Beispieleingabe

```
| Schluesse1  
| 1czRy1LSynZ0pEjezDiyJ4afpdqpa
```

### Beispielausgabe

```
| DasGehtAuchMitZahlen123459876
```

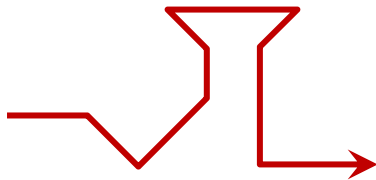
## 8. Pfade

Gegeben ist ein Quadrat mit  $20 \times 20$  positiven ganzen Zahlen. Gesucht werden zwei Pfade (einer von oben nach unten, der andere von links nach rechts), für die die Summe aller Zahlen entlang des Pfades maximal wird. Ein Pfad beginnt in bei einer beliebigen Zahl in der ersten Zeile (oben) bzw. Spalte (links). Weiterhin verläuft ein Pfad monoton horizontal oder vertikal – für den Pfad von links nach rechts bedeutet das, dass der Pfad pro »Schritt« um 1 nach rechts und potentiell um 1 nach oben oder unten läuft. Analog für den vertikalen Pfad.

Ein Pfad kann also ungefähr so aussehen:



Nicht jedoch folgendermaßen (weil in diesem Falle der Pfad rückwärts laufen würde):



oder so (da hier ein Teil des Pfades steiler verläuft als erlaubt):



Letztendlich gesucht wird der Quotient der horizontalen und vertikalen Pfadsummen. Konkret:

$$\text{Ergebnis} = \frac{p_x}{p_y}$$

*Hinweis:* Es gibt einen effizienten Lösungsalgorithmus.

### Eingabe

20 Zeilen mit je 20 positiven ganzen Zahlen, ihrerseits durch je ein Leerzeichen getrennt. Die Zahlen sind klein genug, dass die Pfadsummen den Wertebereich eines vorzeichenbehafteten 32-bit-Ganzzahltyps nicht überschreiten.

### Ausgabe

Quotient der horizontalen und vertikalen maximalen Pfadsumme, mit genau drei Nachkommastellen. Dezimalpunkt oder -komma sind beide gültig.

## Beispieleingabe

```
40 30 42 222 74 265 93 209 115 274 139 177 7 274 12 15 103 36 236 91
294 97 35 272 167 126 18 262 292 48 8 296 85 93 245 4 4 240 276 153
52 8 163 270 242 224 142 72 99 240 199 26 224 198 96 242 295 70 56 247
247 130 216 68 266 142 93 214 30 8 12 163 59 84 40 287 233 65 30 242
283 245 164 53 148 282 73 186 296 3 9 233 184 30 205 221 92 96 5 101
132 228 43 91 228 37 266 140 159 109 10 230 40 114 264 3 266 164 219 283
70 207 218 28 299 78 279 30 179 118 196 138 61 229 110 55 203 73 124 112
16 232 28 187 292 78 194 70 65 203 255 227 176 21 32 225 11 15 92 151
58 237 261 41 213 171 170 111 4 209 99 194 40 108 267 137 179 31 35 221
184 209 264 275 163 268 261 40 198 185 45 188 280 273 54 79 270 286 273 121
208 83 66 156 104 62 188 68 53 194 46 279 280 170 266 148 272 285 178 245
210 130 213 118 165 210 213 66 54 189 166 193 57 213 14 101 143 109 172 101
80 193 287 4 140 65 208 111 8 206 107 285 109 29 211 78 170 247 290 193
148 123 15 164 28 153 222 67 156 165 6 163 114 77 165 17 143 209 278 100
3 102 58 148 82 181 84 29 2 236 231 195 118 278 252 257 179 123 276 287
143 141 254 142 200 243 171 32 164 195 235 260 269 191 190 46 65 166 82 146
69 194 65 220 234 110 45 135 125 208 138 20 233 291 256 162 148 216 247 138
10 53 164 107 2 270 226 227 88 206 193 13 41 130 218 249 76 35 207 91
199 36 207 256 58 215 28 277 234 29 198 148 219 244 136 16 30 258 219 264
183 118 48 218 15 125 279 103 73 8 86 113 9 157 239 273 146 208 50 86
```

## Beispielausgabe

| 0.996

## Beispieleingabe

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 5 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

## Beispielausgabe

| 0.889