

Distriktsmästerskapen i programmering - Uppgiftsöversikt

Problem	Krypto 1: klartext	Krypto 2: xor	Algo 1: konami	Algo 2: rektangelmagi	Mjukvara 1: listtestning	Mjukvara 2: email	AI 1: tictactoe	AI 2: rattafel
Grupp 1	10	6	7	10	6	20	13	60
Grupp 2	15	10	11	10	6	30	19	60
Grupp 3	25	12	12	10	6	25	24	60
Grupp 4	25	19	20	10	6	25	14	
Grupp 5	40	24		10	6			
Grupp 6		33		10	6			
Grupp 7		41		10	6			
Grupp 8		15		10	6			
Grupp 9				10	6			
Grupp 10				10	6			
Summa	115	160	50	100	60	100	70	180

Känd klartext

Problem ID: klartext

När Alice skickar email med sin vän Bob så krypterar hon all text så att inte obehöriga kan läsa den. Totalt använder Alice och Bob fem olika chiffer för att kryptera sina mail.

Din uppgift är att, givet ett krypterat email och vilket chiffer som användes för att kryptera mailet, skriva ut emaillet i klartext. Med klartext menas emaillet som det såg ut innan det krypterades.

Till din hjälp får du för varje chiffer ett antal krypterade email och deras klartext.

Indata

Indata består först av ett tecken 1, 2, 3, 4 eller 5 - vilket chiffer som användes för att kryptera emaillet. Talet är följt av ett mellanslag, och avslutas med det krypterade emaillet på en rad.

Output

Du ska skriva ut klartexten för det krypterade mailet.

Poäng

10 poäng ditt program ska klara av det första chiffret.

15 poäng ditt program ska klara av det andra chiffret.

25 poäng ditt program ska klara av det tredje chiffret.

25 poäng ditt program ska klara av det fjärde chiffret.

40 poäng ditt program ska klara av det femte chiffret.

Sample Input 1

1 19-aqxsmeakkd

Sample Output 1

tjqlfxtdw

Sample Input 2

1 10-clitu

Sample Output 2

mvsde

Sample Input 3

1 21-tfcpk

Sample Output 3

oakfk

Sample Input 4

2 oowysruoqxylxyq

Sample Output 4

oloxwyyqsxryuqo

Sample Input 5

2 xpfmteiggqcxquf

Sample Output 5

xxpqfcmftqeuigg

Sample Input 6

2 txypu

Sample Output 6

tuxpy

Sample Input 7

2 jxwripcv

Sample Output 7

jcxpwvri

Sample Input 8

2 ukhazdnkbyevzw

Sample Output 8

uekvhyawzbdznk

Sample Input 9

3 c b d c b c c b b b d d b c b b b b a c c b c d a

Sample Output 9

a b c b b b c b b c d b a c c c d b b d c b b d c

Sample Input 10

3 c c b d d a a a b c b d b d a b b a c d d b b b a

Sample Output 10

c b d a b b a c a a b b a b d d d a b b d c d b c

Sample Input 11

3 a c c c b b d c d a c a c b d d c a d d d a d c a

Sample Output 11

d c d a d d d a c b c a a d d b a d c c c c b c a

Sample Input 12

3 d a c a b d b b c c a d b d d b c a c a d d c a d

Sample Output 12

c c a a b c b c b d a d d c d d d d b a a a b c d

Sample Input 13

3 d d d c d c c a d b b c b a b a b b b a b d c a b

Sample Output 13

b b a b a d c b a c a d b c b a c b b b c d d d d

Sample Input 14

3 d a a b c d a a b d b b c b d b b c c b d a d d c

Sample Output 14

c b b c b b a d a d d a c d d b b d c b b a c a d

Sample Input 15

3 b b c d a b c a b a a d b d a c c a c c d c a b b

Sample Output 15

c c c a c b c a a b b c b a d d d a b a d b a c b

Sample Input 16

3 d a c a c a a a c a a c a b d c d a c c b c a c a

Sample Output 16

c d c a c c a a a a c c a a b b c d a a a a c c d

Sample Input 17

3 a b a a b d c a b d d b c c b b b b a b b a c b d

Sample Output 17

a b b b b b c d a d b a d c b c b b c d a b b a a

Sample Input 18

4 i s q l e r v n

Sample Output 18

n v r e l q s i

Sample Input 19

4 v h u g n z l v o e h j l

Sample Output 19

g n z v h u l h j l v o e

Sample Input 20

4 c h

Sample Output 20

h c

Sample Input 21

4 l v

Sample Output 21

v l

Sample Input 22

4 y c b u p w a e w

Sample Output 22

u b c y p w e a w

Sample Input 23

4 n z b m z d o

Sample Output 23

n z b m z d o

Sample Input 24

4 e s u q z y l u y m z n z n y

Sample Output 24

e s u q z y l u y m z n z n y

Sample Input 25	Sample Output 25
4 dsjuck	uckdsj
Sample Input 26	Sample Output 26
4 khrvaooowdjehnq	khrvaooowdjehnq
Sample Input 27	Sample Output 27
4 f	f
Sample Input 28	Sample Output 28
4 upau	puaui
Sample Input 29	Sample Output 29
4 dvfbngyidftghz	idftghzdvfbnqy
Sample Input 30	Sample Output 30
4 auaimqkt	tkqmiaua
Sample Input 31	Sample Output 31
5 011100101011011111010101000	ehmgu
Sample Input 32	Sample Output 32
5 0111101000111011110011110111001	msfiqp
Sample Input 33	Sample Output 33
5 110000111111011110000001011110110101110100	omxadxhn
Sample Input 34	Sample Output 34
5 0000001110010110	zzit
Sample Input 35	Sample Output 35
5 0010111110110011110111000	dqvqo
Sample Input 36	Sample Output 36
5 001010010110010101001010101010110110	clcvkht
Sample Input 37	Sample Output 37
5 000000100010011000100111101001010001010	zzuvuvqcc
Sample Input 38	Sample Output 38
5 01110011100000010110110010101101110100100	eadtkxxj
Sample Input 39	Sample Output 39
5 1000001101010110110100	urkxn
Sample Input 40	Sample Output 40
5 1010011001001001000101010011100111000011000101000110	npjugeaont

Sample Input 41**Sample Output 41**

5 111001111100000110010100010101001100110101001001101

iatkcvvgxs

Sample Input 42**Sample Output 42**

5 10011011111001101010

vwig

Sample Input 43**Sample Output 43**

5 11100010010100010011001

bcjp

Sample Input 44**Sample Output 44**

5 101011101011111000010110010010110100000011100101010

hhawjwszeg

Sample Input 45**Sample Output 45**

5 101000011111100100001101110001010

nryjrek

Sample Input 46**Sample Output 46**

5 111000101010001010001000011001010

bkcyjrc

Sample Input 47**Sample Output 47**

5 1010101001000101011111000111101110010001111

gvzhbqpy

Sample Input 48**Sample Output 48**

5 11101001010111010111000001110001001011

qcxezbd

Sample Input 49**Sample Output 49**

5 101001000001111100001100011100000111011101110001001

nuraoafxov

Sample Input 50**Sample Output 50**

5 10101101010111010101110100

hkqln

Sample Input 51**Sample Output 51**

5 10100010111110101000100011100101110111111

nlqssewwy

Sample Input 52**Sample Output 52**

5 1010111100000011100010101100101110100001011

hozblnd

Sample Input 53**Sample Output 53**

5 1110011010111110111100110100

ihqin

Sample Input 54**Sample Output 54**

5 1000010000011100001100010101111101111000101001111

uszaohqbsy

Sample Input 55**Sample Output 55**

5 101010100000100010100010101111101100100101011

gujkcywjh

Sample Input 56**Sample Output 56**

5 1000100101001010111010101010100110

uvshggt

Sample Input 57

5 1000011100110111101

Sample Output 57

uexq

Sample Input 58

5 01111100101110010111010111111

Sample Output 58

mvewhy

Sample Input 59

5 101010001000111011010011001

Sample Output 59

gjfnp

Sample Input 60

5 01010011101001111001010001111001010

Sample Output 60

kfrpsmc

Sample Input 61

5 011110100011100001011100110100001001011101011000

Sample Output 61

msedvnjwhz

Sample Input 62

5 11100001010011001010001010

Sample Output 62

anpsk

Sample Input 63

5 0111010010111101010101001001010000011100000011101

Sample Output 63

fdxkvczez f

Xorxorxor

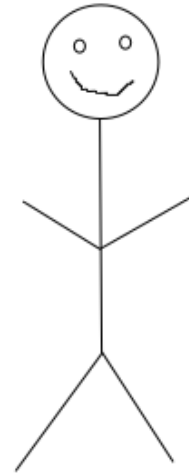
Problem ID: xor

This is Måns

Måns doesn't like that the NSA is reading all his emails, violating his privacy. So Måns encrypts all his emails with a super-secure homemade cipher.

Måns is smart.

Be like Måns.



Måns har kommit på ett briljant chiffer för att skydda sig mot elakingar som försöker läsa hans epost. Hans chiffer fungerar på följande vis. Först använder han teckenkodningen ASCII för att representera varje tecken i sitt mail som ett tal mellan 0-127. Sedan väljer han en nyckel som består av K bytes i intervallet 0-127. Om nyckeln består av bytes $a_0, a_1, a_2, \dots, a_{K-1}$, och hans email består av bytes $b_0, b_1, b_2, \dots$ så består krypteringen av bytes $k_i = b_i \oplus a_{(i \bmod K)}$, där $\oplus$ är en **bitwise XOR** över de 8 bitarna i byten, och $i \bmod K$ är den **sedvanliga modulo-operationen**. Slutligen kodar han varje byte som två tecken i bas 16 (**hexadecimalt**).

Givet Måns krypterade meddelande, bestäm hur det ursprungliga meddelandet såg ut.

Indata

Indata består av ett antal rader - det krypterade meddelandet. Observera att eventuella radbrytningar endast är med för att indata ska bli enklare att skriva, de ska inte tas hänsyn till i problemet.

Varje ursprungligt meddelande består av minst 500 och högst 1050 tecken.

Utdata

Du ska skriva ut, tecken för tecken, Måns ursprungliga meddelande.

Poäng

6 poäng $K = 1$, meddelandet består enbart av små bokstäver $a - z$ och mellanslag.

10 poäng $K = 1$.

12 poäng $K \leq 5$, texten består till 70% av ordet `fisk`.

19 poäng $K \leq 5$, meddelandet består enbart av små bokstäver $a - z$ och mellanslag.

24 poäng $K \leq 5$

[illegible]

Sample Output 1

Fisk. Fisk fisk fisk fisk. Fiskfiskfisk? Fisk! Fisk fisk.

Fisk? Fisk... fisk; fisk fisk. Fisk + fisk = fisk.
Fisk fisk fisk (fisk% fisk) fisk fisk fisk. FISK.
fisk fisk fisk, fisk fisk?

"Fisk fisk fisk", fisk fisk. Fisk fisk? Fisk fisk fisk.

Fiskfiskfiskfisksfisk.
Fiskfiskfiskfisksfisk.
Fiskfiskfiskfisksfisk.
Fiskfiskfiskfisksfisk.
Fiskfiskfiskfisksfisk.
Fiskfiskfiskfisksfisk.

Fisk 1: Fisk fisk, fisk fisk! Fisk :D
Fisk 2: Fisk, fisk fisk :(
Fisk 1: Fisk... Fisk fisk? :)
Fisk 2: Fisk fisk fisk, fisk fisk.
Fisk 1: Fisk fisk, fiskfisk ;)
Fisk 2: Fisk fisk ^^
Fisk 1: Fisk fisk fisk fisk!!
Fisk 2: Fisk fisk fisk <3

Fisk fisk fisk, fisk fisk. Fisk-i-fisk-fisk! Fisk fisk. Fisk, fisk fisk.
Fisk.

Konamikoden

Problem ID: konami

En vanlig fuskkod i många gamla spel är den så kallade *konamikoden*, som består av sekvensen upp upp ned ned vänster höger vänster höger B A.

Du håller på att programmera ett spel, där du vill lägga in ett fusk som aktiveras när man skriver in konamikoden. Dock vill du göra det med en twist - det ska vara tillåtet att trycka på högst K andra knappar mellan din konamikod.

Om $K = 3$ betyder detta att vi får sätta in tre extra knapptryckningar. Alltså skulle upp upp ned **vänster** ned vänster **B B** höger vänster höger B A vara en korrekt konamikod, där de tre extra knapptryckningarna är markerade i fetstil.

Du ska nu skriva ett program som, givet en sekvens av knapptryckningar, avgör det lägsta K -värde som behövs för att konamikoden ska förekomma i sekvensen. Notera att knapptryckningar som sker före den första konamikodstryckningen och efter den sista konamikodstryckningen inte räknas. Detta betyder att för sekvensen **B B vänster** upp upp ned **vänster** ned vänster **B B** höger vänster höger B A **A B upp** ska vi fortfarande svara $K = 3$.

Indata

Indata innehåller en enda rad med N tecken - sekvensen av knapptryckningar. Den kommer ges som en sekvens av bokstäverna U, N, V, H, B, A, som står för knapptryckningarna upp, ned, vänster, höger, B, A.

Det är garanterat att konamikoden finns som en delsekvens av knapptryckningarna.

Utdata

Du ska skriva ut en enda rad med heltalet K som beskrivet i uppgiften.

Poäng

7 poäng $N \leq 11$.

11 poäng $N \leq 100$.

12 poäng $N \leq 3000$.

20 poäng $N \leq 200\,000$.

Sample Input 1

UUNNVHVBHA

Sample Output 1

0

Sample Input 2

UUNVNVBBHVBHA

Sample Output 2

3

Sample Input 3

UUNNVHBVBHBA

Sample Output 3

1

Rektangelmagi

Problem ID: rektangelmagi

En *magisk sekvens* av längd $n + 1$ är en sekvens av tal $a, a + d, a + 2d, \dots, a + nd$ för två rationella tal a och d , t.ex. 2, 5.5, 9, 12.5 eller 5, 5, 5 eller 2, 1, 0, -1, -2.

En *magisk rektangel* av storlek $R \times C$ är en rektangel där varje rad och kolumn är en magisk sekvens.

Givet en rektangel av heltal där vissa av talen är bortsuddade, avgör om det går att fylla i dessa bortsuddade tal så att rektangeln är en magisk rektangel.

Indata

Den första raden i indata innehåller talen R och C , antalet rader och kolumner i den givna rektangeln. Sedan följer R rader med C heltal vardera.

Ett bortsuddat tal representeras som en punkt.

Utdata

Om ingen lösning finns, skriv ut `ej magisk`.

Annars, skriv ut R rader med C kolumner - en magisk rektangel där du tagit indatarektangeln och ersatt de bortsuddade talen.

Rationella tal ska anges på formen N/D , där N och D är högst 100 siffror långa. Observera att det inte ska vara mellanslag mellan talen och divisionstecknet.

Om $D = 1$ kan du skriva N .

Poäng

I de första 9 fallen gäller $1 \leq R, C \leq 6$.

10 poäng alla tal är redan ifyllda.

10 poäng Antingen R eller C är 1.

10 poäng $R = C = 2$

10 poäng varje testfall har en unik lösning, och rektangeln är konstruerad så att det finns en rad eller kolumn med bara ett bortsuddat tal, och när den fylls i finns det återigen en rad eller kolumn med bara ett tal, osv, ända tills hela rektangeln är ifylld.

10 poäng varje testfall har en unik lösning som innehåller enbart heltal.

10 poäng varje testfall har en unik lösning.

10 poäng varje testfall har antingen en unik lösning som innehåller enbart heltal, eller så har det inte en lösning.

10 poäng varje testfall har antingen en unik lösning eller ingen lösning alls.

10 poäng inga ytterligare begränsningar.

10 poäng inga ytterligare begränsningar, men $1 \leq R, C \leq 50$.

Sample Input 1

3	5			
.	.	3	.	5
.	.	.	5	.
.	.	.	.	7

Sample Output 1

1	2	3	4	5
2	3	4	5	6
3	4	5	6	7

Sample Input 2

1	6			
4	.	.	0	.

Sample Output 2

4	8/3	4/3	0	-4/3	-8/3
---	-----	-----	---	------	------

Sample Input 3

```
1 4
1 2 . 2
```

Sample Output 3

```
ej magisk
```

Sample Input 4

```
3 3
1 . .
. 2 .
. . 3
```

Sample Output 4

```
1 0 -1
3 2 1
5 4 3
```

Listtestning

Problem ID: listtestning

Mårten har implementerat en **dubbellänkad lista**. Mårten är nämligen inte så smart. Han vet inte om att det finns färdliga länkade listor i nästan alla standardbibliotek som finns.

Mårten håller inte med dig om att det är dumt - han tycker att sin egna lista är mycket effektivare än den som finns i standardbiblioteket. Det är upp till dig att motbevisa honom, genom att demonstrera att effektivitet inte är allt. Hans lista är nämligen trasig.

Din uppgift är att skriva ett antal testfall som demonstrerat Mårtens trasigheter. Totalt har Mårten gjort 10 försök att skriva en länkad lista, och dina testfall ska skjuta ner så många av Mårtens implementationer som möjligt.

Ett testfall består av en lista av kommandon på följande form:

storlek - fråga vad storleken på listan är.

pop_first - ta bort första elementet i listan.

pop_back - ta bort sista elementet i listan.

add_first X - lägg till heltalet $-1000 \leq X \leq 1000$ först i listan.

add_back X - lägg till heltalet $-1000 \leq X \leq 1000$ sist i listan.

add X Y - lägg till heltalet $-1000 \leq X \leq 1000$ på plats Y i listan.

remove Y - ta bort elementet på plats Y i listan.

clear - ta bort alla element i listan.

Positioner i listan är noll-indexerade.

Mellan testfall ska du skriva ut en rad med tre bindestreck: ---.

Indata

Problemet har ingen indata.

Utdata

Du ska skriva ut ett antal rader med dina testfall. Du får skriva ut max 1000 rader.

Poängsättning

Totalt har Mårten kodat 10 trasiga implementationer. För varje implementation som dina testfall dödar får du 6 poäng.

Sample Input 1	Sample Output 1
	storlek clear clear clear --- clear storlek storlek clear add 1000 0

Email-parsing

Problem ID: email

Nyckelorden **MÅSTE**, **MÅSTE INTE**, **KRÄVS**, **SKALL**, **SKALL INTE**, **BÖR**, **BÖR INTE**, **REKOMMENDERAS**, **FÅR**, **FRIVILLIGT** i denna uppgiftslydelse skall tolkas som beskriver i RFC 2119.

För att lösa denna uppgift **REKOMMENDERAS** det att läsa följande specifikationer:

- RFC 20: <https://www.ietf.org/rfc/rfc20.txt>
- RFC 1866: <https://www.ietf.org/rfc/rfc1866.txt>
- RFC 2045: <https://www.ietf.org/rfc/rfc2045.txt>
- RFC 2046: <https://www.ietf.org/rfc/rfc2046.txt>
- RFC 2119: <https://www.ietf.org/rfc/rfc2119.txt>
- RFC 2616: <https://www.ietf.org/rfc/rfc2616.txt>
- RFC 2822: <https://www.ietf.org/rfc/rfc2822.txt>
- RFC 3676: <https://www.ietf.org/rfc/rfc3676.txt>

Du kommer få email i formatet som beskrivs av RFC 2822. Din uppgift är att hitta en föredragen del i detta email, vars innehåll är ett MIME-meddelande, som specificerat i RFC 2045 och RFC 2046. Texten i denna del **SKALL** du sedan skriva ut.

Meddelandet kommer alltid vara av typen `multipart/alternative`. Den föredragna delen i meddelandet är antingen av typen `text/html`, som beskrivs i RFC 1866, eller typen `text/plain`, som beskrivs i RFC 3676. Teckenkodningen kommer att vara ASCII, som beskriven i RFC 20.

För enkelhets skull kommer varje rad att termineras med en enkel LF i stället för en CRLF som i vanlig HTTP 1.1 (beskriven i RFC 2616).

Till skillnad från en riktig email-klient **FÅR** du anta att saker och ting kommer att vara snälla. T.ex. kommer HTML-en i en `text/html`-del alltid validera.

Indata

Indata innehåller ett email, se exempel nedan.

Utdata

Du **SKALL** skriva ut den extraherade texten och inget annat. Extra nyrader i början/slutet är **FRIVILLIGT**.

Poäng

20 poäng det kommer bara finnas en enda multipart-del, och den är av typen `"text/plain"`.

30 poäng den föredragna delen kommer att vara av typ `"text/plain"`.

25 poäng den föredragna delen kommer att vara av typ `"text/plain"`, och kommer att kodas med `"Content-Transfer-Encoding: quoted-printable"`.

25 poäng den föredragna delen kommer att vara av typ `"text/html"`. Alla `<HTML-taggar>` och `&entiteter`; ska tas bort.

Endast i den tredje gruppen kommer kodningen vara `"Content-Transfer-Encoding: quoted-printable"`. I de övriga kommer den att vara `"7bit"`.

Sample Input 1

```
MIME-Version: 1.0
Received: by 10.112.236.98 with HTTP; Thu, 28 Jan 2016 10:57:40 -0800 (PST)
Date: Thu, 28 Jan 2016 19:57:40 +0100
Delivered-To: simon.lindholm10@gmail.com
Message-ID: <CAK=Tff0+3K4tXGvYh+VG-qP6Ldz+q-+jaZW2Q01dkL4fDQxGkw@mail.gmail.com>
Subject: test
From: Simon Lindholm <simon.lindholm10@gmail.com>
To: simonlindholm@fastmail.fm
Content-Type: multipart/alternative; boundary=001a11c269f42ab6b0052a697e72

--001a11c269f42ab6b0052a697e72
Content-Type: text/plain; charset=UTF-8

a test

--001a11c269f42ab6b0052a697e72--
```

Sample Output 1

```
a test
```

Sample Input 2

```
Delivered-To: simon.lindholm10@gmail.com
Received: by 10.112.155.195 with SMTP id vy3csp710980lbb;
      Thu, 28 Jan 2016 14:08:38 -0800 (PST)
Return-Path: <simonlindholm@fastmail.fm>
Received: from out3-smtp.messagingengine.com (out3-smtp.messagingengine.com. [66.111.4.27]
      by mx.google.com with ESMTPS id wb3si19310522pab.114.2016.01.28.14.08.37
      for <simon.lindholm10@gmail.com>
      (version=TLS1_2 cipher=ECDHE-RSA-AES128-GCM-SHA256 bits=128/128);
      Thu, 28 Jan 2016 14:08:37 -0800 (PST)
Received: from compute2.internal (compute2.nyi.internal [10.202.2.42])
      by mailout.nyi.internal (Postfix) with ESMTP id 965F620767
      for <simon.lindholm10@gmail.com>; Thu, 28 Jan 2016 17:08:34 -0500 (EST)
Received: from web5 ([10.202.2.215])
      by compute2.internal (MEProxy); Thu, 28 Jan 2016 17:08:34 -0500
Received: by web5.nyi.internal (Postfix, from userid 99)
      id 26696A70216; Thu, 28 Jan 2016 17:08:34 -0500 (EST)
Message-Id: <1454018914.1678085.505620754.5295DB0C@webmail.messagingengine.com>
From: Simon Lindholm <simonlindholm@fastmail.fm>
To: Simon Lindholm <simon.lindholm10@gmail.com>
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Type: multipart/alternative; boundary="_-----=_145401891416780850";
      charset="utf-8"
Subject: Re: test
Date: Thu, 28 Jan 2016 23:08:34 +0100
In-Reply-To: <CAK=Tff0+3K4tXGvYh+VG-qP6Ldz+q-+jaZW2Q01dkL4fDQxGkw@mail.gmail.com>
References: <CAK=Tff0+3K4tXGvYh+VG-qP6Ldz+q-+jaZW2Q01dkL4fDQxGkw@mail.gmail.com>

This is a multi-part message in MIME format.

--_-----=_145401891416780850
Content-Transfer-Encoding: 7bit
Content-Type: text/plain

a reply

On Thu, 28 Jan 2016, at 07:57 PM, Simon Lindholm wrote:
> a test

--_-----=_145401891416780850
Content-Transfer-Encoding: 7bit
Content-Type: text/html

<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body><div>a reply<br></div>
<div>&nbsp;</div>
<div>On Thu, 28 Jan 2016, at 07:57 PM, Simon Lindholm wrote:<br></div>
<blockquote type="cite"><div dir="ltr">a test<br></div>
</blockquote><div>&nbsp;</div>
</body>
</html>

--_-----=_145401891416780850--
```

Sample Output 2

a reply

On Thu, 28 Jan 2016, at 07:57 PM, Simon Lindholm wrote:
a test

3-i-rad

Problem ID: tictactoe

Mårten är extremt duktig på 3-i-rad. Så duktig att du aldrig lyckas slå honom! Skriv ett datorprogram som spelar 3-i-rad åt dig, så du kanske har en chans.

Om du inte vet vad 3-i-rad är, kan du läsa om hur spelet fungerar [på Wikipedia](#).

Indata och utdata

Detta problem är interaktivt. Du ska först läsa en rad som innehåller antingen `first` or `second` - detta säger om du ska dra först eller inte.

Varje gång du ska dra ska du läsa in ett 3-i-rad-bräde med draget Mårten gjorde (eller ett blankt bräde för det allra första draget), och sedan skriva ut ett 3-i-rad-bräde med draget du gjorde.

Om du skriver ut ett bräde där du vinner eller spelar oavgjort ska ditt program avslutas. Om du läser in ett bräde där Mårten vunnit eller spelat oavgjort ska ditt program avslutas.

Ett bräde ska skrivas ut och läsas in som 3 rader med 3 tecken vardera - antingen `.` för tom ruta, `o` för Mårtens pjäser, eller `x` för dina pjäser (obs: små bokstäver).

Exempel

I exemplet nedan spelar Mårten först. Mårtens utskrift är i fetstil.

second

```
..o
...
...
..o
.x.
...
.oo
.x.
...
xoo
.x.
...
xoo
.xo
...
xoo
.xo
...
..x
```

Efter det sista draget vann du mot Mårten, och ska därmed avsluta ditt program.

Poäng

13 poäng du ska spela ett giltigt spel.

19 poäng du ska aldrig förlora mot Mårten.

24 poäng du ska alltid vinna ett spel om det går.

14 poäng du ska aldrig förlora mot Mårten, och du ska alltid vinna ett spel om det går.

Rätta fel

Problem ID: rattafel

Givet en text på engelska, där vissa tecken har blivit ersatta med #, rätta texten genom att fylla i vilka tecken som skulle stått där egentligen.

Indata

Indata består av en enda rad med tecken a-z, A-Z, 0-9, skiljetecknen ., -, ;, !, ?, " ' , samt blanksteg. Vissa av dessa tecken har blivit ersatta av #. Ingen testdatafil består av mer än 500 000 tecken.

Indatafilen kan även innehålla nyrader, men inget av dessa kommer ersättas av #.

Utdata

Du ska skriva ut indatatexten, där varje förekomst av # har ersatts med det ursprungliga tecknet på den positionen. Om du ändrar ett annat tecken än # kommer du få fel på det testfallet.

Alla tecken kan ha blivit ersatta, inklusive mellanslag och skiljetecken. Om du ska rätta en bokstav, måste du även ha rätt på stor och liten bokstav.

Det spelar ingen roll hur du skriver ut nyrader - dessa kommer ignoreras vid rättning (och behöver alltså inte överensstämma med nyraderna i indata).

Poäng

Det finns tre testfall ditt program kommer köras på.

1. I ett testfall värt 60 poäng kommer högst en bokstav per ord att ersättas. Inga mellanslag eller skiljetecken kommer ersättas.
2. I ett testfall värt 60 poäng kommer 10% slumpmässiga tecken bytas ut.
3. I ett testfall värt 60 poäng kommer 30% slumpmässiga tecken bytas ut.

Detta problem kan alltså ge totalt 180 poäng.

Om du på ett visst testfall, som ger P poäng, rättar M fel och det totalt finns N fel får du

$$P \cdot \frac{M}{N}$$

poäng, avrundat nedåt.

Din totala poäng är summan av poängen för de olika testfallen.

Sample Input 1

```
MAR#EY was dead: to begin with. The#e is no doubt whate##r
about#that. T#e register of his bu#ial was signe# by the
clerg#man, the cl#rk, the u#dertaker, an# the chief mou#ner.
Scrooge s#gned it: and Scr#oge's name was g#od upon 'Ch#nge,
for anyt#ing he chose to#put his h#nd#to. Old Ma##ey wa# ##
dead as#a door-nail. #Mind! I d#n't mean##o say that I#know,
of my ow##k#owledge, #hat there is #articul#r#y dead ab##t a
door-nail. I#might have bee# inclined, myself, to #egard a
coff#n-nail as the ##adest piec# of ironmonge#y in the
t#ade. But the#wisdom of#our ancestors is#in the simile; a#d
my unhallowed#hands shall n#t disturb i#, or the Country's
done fo## You will therefor# permit me to re#eat,
emphat#cally, that Marle# was as dead #s a doo#-#ail.
```

Sample Output 1

MARLEY was dead: to begin with. There is no doubt whatever about that. The register of his burial was signed by the clergyman, the clerk, the undertaker, and the chief mourner. Scrooge signed it: and Scrooge's name was good upon 'Change, for anything he chose to put his hand to. Old Marley was as dead as a door-nail. Mind! I don't mean to say that I know, of my own knowledge, what there is particularly dead about a door-nail. I might have been inclined, myself, to regard a coffin-nail as the deadest piece of ironmongery in the trade. But the wisdom of our ancestors is in the simile; and my unhallowed hands shall not disturb it, or the Country's done for. You will therefore permit me to repeat, emphatically, that Marley was as dead as a door-nail.