

パソコン甲子園2005

全国高等学校パソコンコンクール
プログラミング部門 予選大会問題

平成 17 年 9 月 3 日(土)午前 10 時～正午 (時間厳守)



全国高等学校パソコンコンクール実行委員会

解答にあたっての注意点

1 解答方法

2 作成するプログラム

A 2

10

A10.c A10.exe

01

BASIC

3 データ型

4 入力

1

2

1

Enter

2

Enter

.CSV

1

1,2,3

4,5

1

2

EOF

data+

+

c:\pckosien¥

+

5 出力

=

1

2

3

6 不正解となる例

ファイル名のつけ方が違う

ファイルがそろっていない

BASIC

A01.c

A01.exe

出力形式が違う

1 1

1 1

1

終了しない

問題01 四角すいの表面積(6点)

1 辺が x の正方形を底面とする、高さ h の四角すいの表面積 S を出力して終了するプログラムを作成して下さい。ただし、頂点と底面の中心を結ぶ線分は底面と直交しているとします。また、 x 、 h は 100 以下の正の整数とします。

入力

x (整数)

h (整数)

出力

S (実数)

入力例1

6

4

出力例1

96.000000

入力例2

7

9

出力例2

184.192459

問題02 ビデオテープの残り時間(6点)

標準録画で 120 分のビデオテープがあります。テープを完全に巻き戻した状態でビデオデッキのカウンタを 00:00:00 にし、標準録画モードで録画したところ、あるカウンタ値になりました。このカウンタ値(時、分、秒)を入力し、残りのテープの長さ(録画可能時間)を求め、時:分:秒の形式で出力して終了するプログラムを作成して下さい。

ただし、2 時間(120 分)以内の入力とします。なお、テープ残量は標準録画モードと 3 倍録画モードの場合の 2通りを計算し、出力例のように時、分、秒とも 2 桁ずつ出力します。また”05”のように 10 の位が 0 の場合は、”0”を必ずつけてください。

入力

時(整数)

分(整数)

秒(整数)

出力

テープの残りを標準録画した場合の録画可能時間 時:分:秒(半角コロン区切り)

テープの残りを3倍録画した場合の録画可能時間 時:分:秒(半角コロン区切り)

入力例

1

30

0

出力例

00:30:00

01:30:00

問題03 BMI(6点)

肥満は多くの成人病の原因として挙げられています。過去においては、一部の例外を除けば、高校生には無縁なものでした。しかし、過度の受験勉強等のために運動不足となり、あるいはストレスによる過食症となることが、非現実的なこととはいえなくなっています。高校生にとっても十分関心を持たねばならない問題になるかもしれません。

そこで、あなたは、保健室の先生の助手となって、生徒のデータから肥満の疑いのある生徒を探し出すプログラムを作成することになりました。

方法は BMI(Body Mass Index)という数値を算出する方法です。BMI は次の式で与えられます。

$$BMI = \frac{(\text{kg})}{(\text{m})^2}$$

BMI=22 が標準的で、25 以上だと肥満の疑いがあります。ファイルに格納されている、各々の生徒の体重、身長の情報から BMI を算出し、25 以上の生徒の学籍番号を画面に出力して終了するプログラムを作成してください。なお、肥満の疑いのある生徒がない場合は、”該当なし”と出力して終了するようにしてください。

入力データのファイル名は data03.csv とします。

入力

学籍番号(整数),体重(実数),身長(実数) (1 件目のデータ)

学籍番号(整数),体重(実数),身長(実数) (2 件目のデータ)

出力

BMI が 25 以上の生徒の学籍番号(複数ある場合、改行してすべて表示)または 該当なし

入力例

1001, 50.0, 1.60

1002, 60.0, 1.70

1003, 70.0, 1.80

1004, 80.0, 1.70

1005, 90.0, 1.60

出力例

1004

1005

蔵を片付けていたら、先祖の宝のありかへの行き方を記した古文書が見つかりました。この古文書にはおおよそ次のような事が書かれていました。

1. まず、町外れの井戸から、真東に 1m の地点に立ち、まっすぐ井戸の方向を向け。
2. 右回りに 90 度向きを変え、1m 直進したら、まっすぐ井戸の方向を向け。
3. 右回りに 90 度向きを変え、1m 直進したら、まっすぐ井戸の方向を向け。
4. //
5. //
6. :

2 行目以降は、全く同じことが書かれていました。あなたは、宝を探そうと思いましたが、厄介なことに気がつきました。昔と違い建物が邪魔をしていて、まっすぐ井戸の方向をむこうにも井戸が見えなかったり 1m 直進しようにも直進できなかったりします。さらに、この古文書ときたら、1000 行近くあり、古文書どおりの作業をすることは、かなりの時間と体力が要ります。しかしながら、幸運なことに、あなたはコンピュータを利用することができます。

古文書に書かれている行数 n を入力して、宝の位置(最後に立った場所)を出力して終了するプログラムを作成してください。ただし、 n は 2 以上 1000 以下の正の整数とします。

入力

古文書の行数 n (整数)

出力

町外れの井戸から東に x (m)、北に y (m) が宝の位置(最後に立った場所)として、

x (実数)

y (実数)

小数点以下第 2 位まで出力してください。小数点以下第 3 位は切り上げ、切り捨ていずれでも正解とします。

入力例 1

3

出力例 1

0.29

1.71

入力例2

6

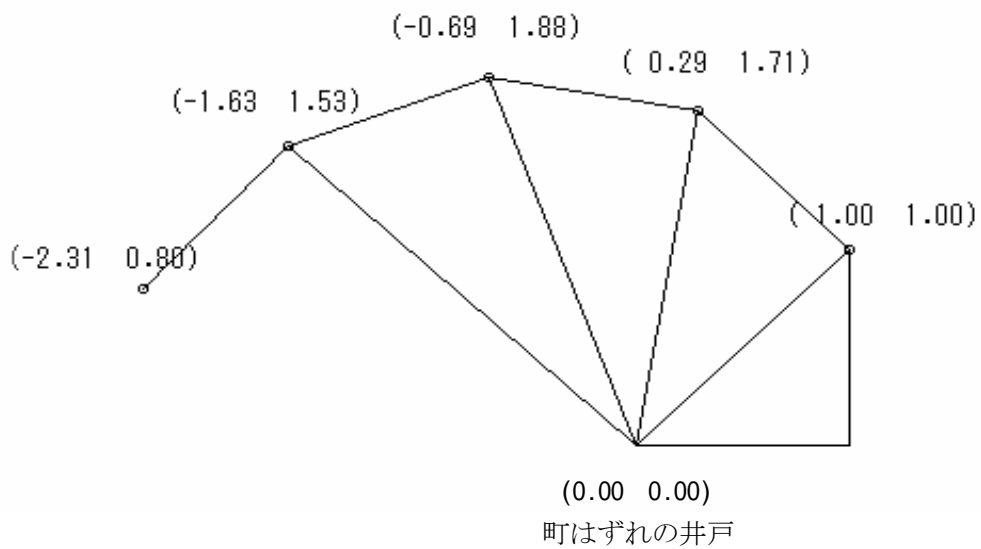
出力例2

-2 31

0.80

参考

$n=2, 3, 4, 5, 6$ の場合の宝の位置



問題05 ランレングス(6点)

文字列が連続した場合、ある規則で文字を置き換え文字列を短くすることができます。たとえば、AAAA という文字列の場合、@4A と表現すれば 1 文字分圧縮されます。この規則で圧縮された文字列を入力してもとの文字列に復元して終了するプログラムを作成してください。ただし、復元した文字列に@文字は出現しないものとします。

原文の文字列は半角の英大文字、英小文字、数字、記号であり 100 文字以内、連続する文字は 9 文字以内です。

入力

文字列

出力

復元した文字列

入力例1

ab@5C1@8050

出力例1

abCCCCC10000000050

入力例2

@9+1=1@90

出力例2

99999999+1=1000000000

問題06 魔方陣(7点)

$n \times n$ の正方形のマス目の中に数字の1から n^2 が一つずつ入っていて、縦のどの列のマス目の和も、横のどの列のマス目の和も対角線のマス目の和も等しいものを魔方陣といいます。

一辺のマスの個数が奇数の魔方陣の作り方には以下の方法があります。

1. 中央のマス目のちょうど一つ下のマス目に1をいれる。
2. 次の数字を右斜め下のマス目にいれる。

ただし、数字をいれようとしたマス目が正方形からはみ出している場合、既に数字が埋まっている場合は以下の方法に従って数字を入れるマス目を探す。

- ・ 右にはみ出した場合には、同じ行の左はしに、左にはみ出した場合には、同じ行の右はしに、下にはみ出した場合には、同じ列の一番上に数字をいれる。
- ・ 数字をいれようとしたマス目が埋まっているときには、その埋まっているマス目の左斜め下のマス目にいれる。

3. すべてのマス目が埋まるまで2を繰り返す。

この方法に従って、一辺のマス目の個数 n を入力として、その大きさの魔方陣を出力して終了するプログラムを作成して下さい。ただし、 n は3以上15以下の奇数とします。マス目に入る各数字は右詰5桁で出力して下さい。

入力

n (正の整数)

出力

$n \times n$ の魔方陣

入力例

5

出力例

11	24	7	20	3
4	12	25	8	16
17	5	13	21	9
10	18	1	14	22
23	6	19	2	15

問題07 凸多角形の面積(7点)

凸 n 角形(すべての頂点の内角が180度未満である多角形、要するにへこんでいない多角形のこと)の頂点の座標をファイルから読んで、その面積を画面に出力して終了するプログラムを作成してください。頂点には辺のつながりの順序に従って頂点1、頂点2、頂点3、・・・頂点 n という名前がついています。

ただし、 n は3以上20以下とします。また、次の、三角形の3辺の長さ a, b, c から面積 S を求める公式を使ってもかまいません。

$$z = \frac{a+b+c}{2}, S = \sqrt{z(z-a)(z-b)(z-c)}$$

入力データのファイル名は data07.csv とします。

入力

頂点1の x 座標, 頂点1の y 座標(全て実数)

頂点2の x 座標, 頂点2の y 座標(全て実数)

・
・
・

頂点 n の x 座標, 頂点 n の y 座標(全て実数)

出力

S (実数)

入力例

0.0 0.0

0.0 1.0

1.0 1.0

2.0 0.0

1.0 -1.0

出力例

2.500000

問題08 3乗根(7点)

$x^3 = q$ の解は漸化式 $x_{n+1} = x_n - \frac{x_n^3 - q}{3x_n^2}$ を計算していくことで近似的に求めることができます。 x_1 に正の数 $\frac{q}{2}$

をいれ、 $x_2 = x_1 - \frac{x_1^3 - q}{3x_1^2}$ 、 $x_3 = x_2 - \frac{x_2^3 - q}{3x_2^2}$ 、 \dots と計算します。

この計算をしながら、 $|x^3 - q|$ の値が、十分小さくなったところで、計算をやめ、最後に計算した x_n を

$x^3 = q$ の近似解とします。

この方法に従って、入力された正の整数 q に対し、 q の3乗根の近似値を出力して終了するプログラムを作成してください。ただし、「十分小さくなった」という判定は $|x^3 - q| < 0.00001q$ を用いてください。

入力

q (整数)

出力

x (実数)

入力例

15

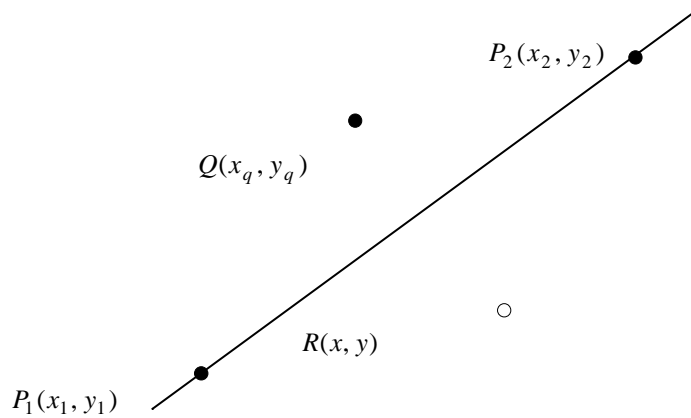
出力例

2.466212

問題09 平面上の対称の位置(7点)

平面上の異なる3点 $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $Q(x_q, y_q)$ の座標の組をファイルから読み込んで、それぞれの組について、点 P_1 点 P_2 を通る直線を対称軸として点 Q と線対称の位置にある点 $R(x, y)$ を出力して終了するプログラムを作成してください。なお、点 Q は、その対称軸上にないものとします。

入力データのファイル名は data09.csv とします。



入力

$x_1, y_1, x_2, y_2, x_q, y_q$ (1 件目のデータ: 全て実数)

: (2 件目のデータ: 全て実数)

出力

$x y$ (1 件目のデータに対する出力: 全て実数空白区切り)

: (2 件目のデータに対する出力: 全て実数空白区切り)

入力例

1.0 0.0 0.0 -1.0 0.0 0.0 1.0 1.0

1.0 0.0 0.0 0.0 -1.0 3.0 0.0 0.0

0.0 1.0 0.0 0.0 -1.0 1.0 1.0

出力例

1.000000 -1.000000

1.000000 2.000000

-1.000000 1.000000

問題10 メリーゴーランド(7点)

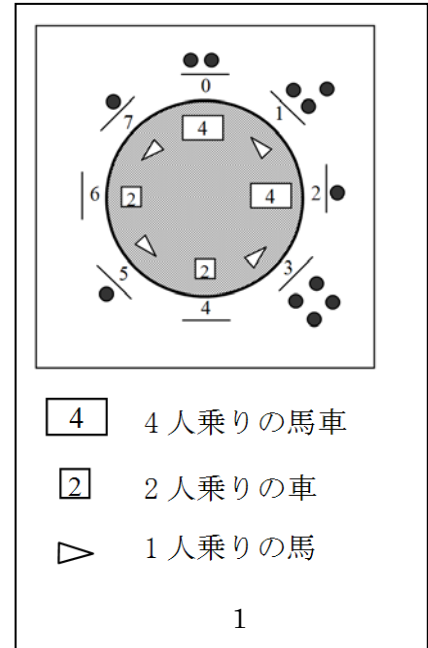


遊園地にあるメリーゴーランドはご存じでしょう。大きな円盤の上に馬や馬車などの乗り物が固定されていて、円盤が回転すると同時に乗り物が上下に揺れる、定番の遊具です。ある遊園地のメリーゴーランドは、4人乗りの馬車が2台、2人乗りの車2台、1人乗りの馬が4台、計8台の乗り物が図1のような順序で備えられています。そして、遊園地においてのお客様は、図1に示す乗り場0~7のどこかで待つようになっています。

この遊園地のメリーゴーランドは、かならず乗り物が乗り場にぴったりと合う位置に停止します。そして、0~7のそれぞれで待っているお客さまは、目の前にとまった乗り物に乗ることになっています。急いで他の乗り場へ移動してそこから乗るということはできません。効率よく、お客さまにたのしんでいただくためには、メリーゴーランドの停止する位置をうまく調整して、乗れないお客さまをできるだけ少なくするようにしなければなりません。

乗り場0~7で待っているお客さまの人数をファイルから読み込んで、それぞれについてどの位置にどの乗り物が来るように止めれば乗れないお客さまが最も少なくなるかを出力して終了するプログラムを作成してください。なお、乗れなくなるお客さまが最小となるとめ方が何通りもある場合は、そのうちの一つだけ出力するようにしてください。

入力データのファイル名は data10.csv とします。



入力

乗り場 0,1,2,...,7 で待っているお客様の人数をそれぞれ p_0, p_1, \dots, p_7 とすると、

$p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7$ (1 件目のデータ: 全て整数)

: (2 件目のデータ: 全て整数)

出力

メリーゴーランドの乗り物の馬車を4、車を2、馬を1で表すこととします。乗り場 0,1,2,...,7 にとめる乗り物を、それぞれ c_0, c_1, \dots, c_7 とすると、

$c_0 c_1 c_2 c_3 c_4 c_5 c_6 c_7$ (1 件目のデータに対する出力: 1 又は 2 又は 4 空白区切り)

: (2 件目のデータに対する出力: 1 又は 2 又は 4 空白区切り)

: :

入力例

2 3 1 4 0 1 0 1

4 2 3 2 2 2 1 1

出力例

1 4 1 4 1 2 1 2

4 1 4 1 2 1 2 1

西暦で表された年月日を、元号を用いた和暦に変換し、その年月日を出力して終了するプログラムを作成してください。入力は例に示すように3つの整数であり、順に西暦年、月、日です。これを、出力例に示すように変換してください。なお、明治以前の年月日が入力された場合は「明治以前」と表示してください。

各年号の最初の年は「元年」ではなく「1年」と出力することとします。

元号	期	間
明治	1868. 9. 8	～ 1912. 7. 29
大正	1912. 7. 30	～ 1926. 12. 24
昭和	1926. 12. 25	～ 1989. 1. 7
平成	1989. 1. 8	～

入力

年(整数)

月(整数)

日(整数)

出力

元号 年月日

入力例1

2005

9

3

出力例1

17 9 3

入力例2

1868

12

2

出力例2

1 12 2

入力例3

1868

9

7

出力例3

問題12 サーチエンジン(7点)

インターネットの検索エンジン、例えば、Google などでは、世界中のウェブページを自動で収録して分類し、巨大なデータベースを作成します。また、ユーザが入力した検索キーワードを解析して、データベース検索のための問い合わせ文を作成します。

いずれの場合も、効率的な検索を実現するために複雑な処理を行っていますが、とりあえずの基本は全て文章からの単語の切り出しです。

ということで、文章からの単語の切り出しに挑戦してください。今回は以下の通り、単語区切りが明確な英語の文章を対象とします。

- ・ 対象となる文章 : 改行を含まない 1024 文字以下の英語の文章
- ・ 区 切 り 文 字 : いずれも半角で空白、ピリオド、カンマのみ
- ・ 切 り 出 す 単 語 : 3 から 6 文字の単語(2 文字以下や 7 文字以上の単語は無視)

入力データのファイル名は data12.txt とします。

入力

区切り文字及び英数字で構成される英文1行(すべて半角)

出力

空白文字1文字(半角)で区切られた単語

入力例1

Rain, rain, go to Spain.

出力例1

Rain rain Spain

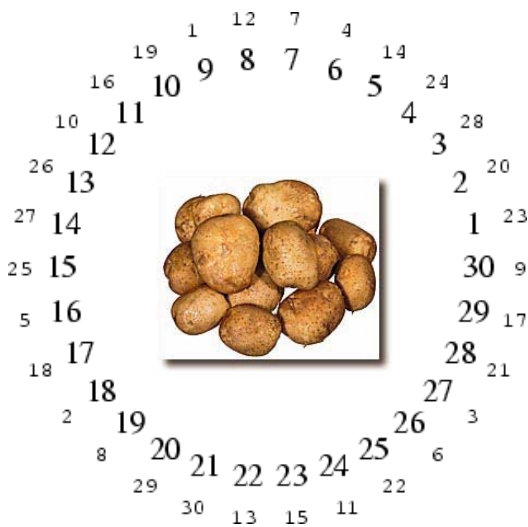
入力例2

Win today's preliminary contest and be qualified to visit University of Aizu.

出力例2

Win and visit Aizu

問題13 ヨセフのおいも(7点)



昔、ヨセフのおイモというゲームがありました。 n 人が参加しているとしましょう。参加者は中心を向いて円陣を組み、1から順番に番号が振られます。アツアツのおイモがひとつ、参加者 n (左の図は $n=30$ の場合で、内側の大きい数字の30)に渡されます。おイモを渡された参加者は右隣の参加者にそのおイモを渡します。 m 回目に渡された人は右隣の人に渡して円陣から抜けます(左の図では $m=9$ の場合を表しています)。 m 回渡す毎に一人ずつぬけ、最後に残った人が勝者となり、おイモをいただきます。 n, m が決まってから、実際におイモを渡し始める前にどこにいたら勝てるかわかるといいですね。

左の図は30人の参加者で9人ごとに抜けるというルールでこのゲームをした場合を書き表しています。内側の大きい数字が参加者に振られた番号、外側の小さい数字が抜ける順番です。それによると、9,18,27,6,16,26という順番で円陣から抜け出し、最後には21が残ることになります。すなわち21が勝者となります(小さい数字が30になっています)。ゲーム参加者数 n と円陣から抜け出す参加者の間隔 m を入力し、勝者の番号を出力して終了するプログラムを作成してください。ただし、 $m < n < 1000$ とします。

入力

ゲーム参加者数 n (整数)

円陣から抜け出す参加者の間隔 m (整数)

出力

勝者となりおイモをいただく人の番号(整数)

入力例1

41

3

出力例1

31

入力例2

30

9

出力例2

21

文久2(1862)年、会津の殿様が京都守護職を命ぜられました。京都守護職とは治安の悪化した幕末の京都を守る大切な役目です。幕府や他の藩と分担して市中をパトロールしなければなりません。ところがいざ分担ルートを決める段になって、家臣の中でも有名な頑固者の古老から以下のような注文がきました。

- 一、 武士たる者、スタート地点とゴール地点は厳に守らねばならぬ。
- 一、 一度のパトロールで、二度・三度同じ道を辿るは、武士の面目に関わる。
- 一、 ただし、交差点は二度以上通るもやむなし。

大変なことになりました。殿様といえどもこの家臣の言い分を無視するわけにはいきません。分担ルートの選択によっては、「武士の面目が立たない」ということになってしまいます。

ということで、スタート地点、ゴール地点、交差点の情報を入力として、上の三つの条件を満たすかどうかを判定するプログラムを作って、殿様に献上してください。

スタート地点を1、ゴール地点を2、その他の交差点を3以上の整数で表します。1つの道路は、その道が結ぶ1組の交差点番号で表します。なお、交差点の数は100以下とし、全ての交差点からスタート地点およびゴール地点への経路がそれぞれ一つ以上あるものとします。

入力データのファイル名は data14.csv とします。

入力

a, b 交差点 a と b とをつなぐ道路 (a, b とも自然数、カンマ区切り)

c, d

:

:

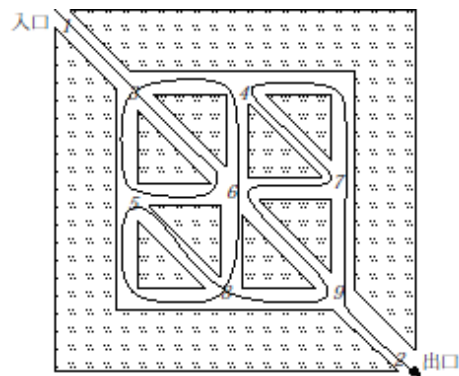
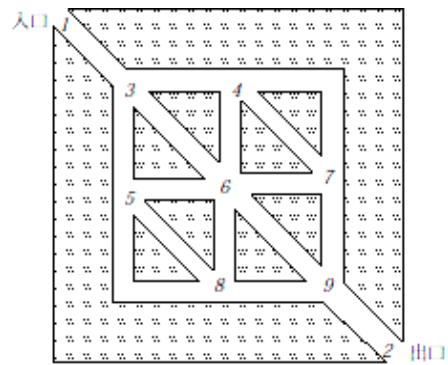
出力

OK 武士の面目が立つ場合(三つの条件を満たす場合)

NG それ以外の場合(三つの条件を満たさない場合)

入力例1

- 1, 3
- 3, 4
- 3, 5
- 3, 6
- 4, 6
- 4, 7
- 4, 7
- 5, 6
- 6, 7
- 5, 8
- 5, 8
- 6, 8
- 6, 9
- 7, 9
- 8, 9
- 9, 2

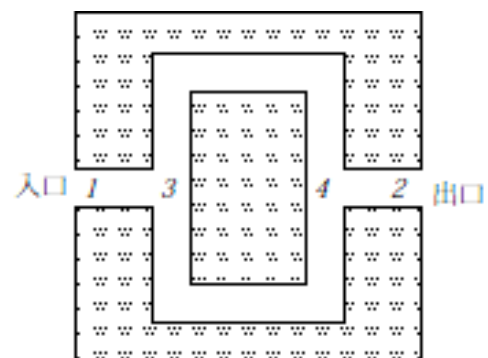


出力例1

OK

入力例2

- 1, 3
- 3, 4
- 3, 4
- 4, 2



出力例2

NG

博士：ピーター君、ついにやったよ。

ピーター：またですか。今度はどんなくだらない発明ですか。

博士：ついに数式を計算機で処理する画期的な方法を思いついたんだ。この表をみてごらん。

通常の数式	博士の「画期的な」記法
$1 + 2$	1 2 +
$3 * 4 + 7$	3 4 * 7 +
$10 / (2 - 12)$	10 2 12 - /
$(3 - 4) * (7 + 2 * 3)$	3 4 - 7 2 3 * + *

ピーター：はあ。

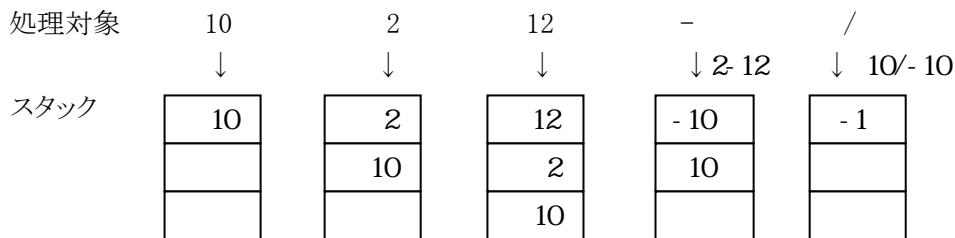
博士：ふっふっふ。これだけでは、未熟者の君には何のことだかわからないだろうねえ。ここからが肝心なんじゃ。

ピーター：っていうか・・・。

博士：計算機にはスタックというデータ構造があることは君も知っているね。ほれ、「先入れ後出し」のあれじゃよ。

ピーター：はい。知ってますが、あの・・・。

博士：この画期的な記法はあのスタックを使うんじゃ。例えばこの 10 2 12 - / だが、次のように処理する。



博士：どうじゃな。括弧も演算子の優先順位も気にする必要がないじゃろう？語順も「10を2から12を引いたもので割る。」となり、何となく彼の極東の島国の言葉、日本語と似ておるじゃろうて。

この画期的な発明さえあれば、我が研究室は安泰じゃて。ファファファ。

ピーター：っていうか博士。これって日本にいたとき会津大学の基礎コースで習いましたよ。「逆ポーランド記法」とかいって、みんな簡単にプログラムしてました。

博士：・・・。

ということで、ピーター君に変わって博士に、このプログラムを教える事になりました。「逆ポーランド記法」で書かれた数式を入力とし、計算結果を出力し終了するプログラムを作成してください。

入力

逆ポーランド記法による数式(数と演算記号が空白文字1文字(半角)で区切られた 80 文字以内の文字列。ただし、数は 10 個以内で全て正の整数)

出力

計算結果(実数)

入力例1

10 2 12 - /

出力例1

-1.000000

入力例2

3 4 - 7 2 3 * + *

出力例2

-13.000000

博士：ピーター君、ついにやったよ。

ピーター：どうしました、デビッド博士？またくだらない発明ですか？

博士：

文字	符号	文字	符号	文字	符号	文字	符号
(空白)	101	C	0101	K	0110	S	00110
'	000000	D	0001	L	00100	T	00111
,	000011	E	110	M	10011001	U	10011100
-	10010001	F	01001	N	10011110	V	10011101
.	010001	G	10011011	O	00101	W	000010
?	000001	H	010000	P	111	X	10010010
A	100101	I	0111	Q	10011111	Y	10010011
B	10011010	J	0011000	R	1000	Z	10010000

ピーター：なんですか？この表は。

博士：いいから、言う通りにしてみなさい。まず、お前の名前を紙に書いてごらん。

ピーター：はい“PETER POTTER” っと。

博士：そうしたら、1文字ずつ、この表の「符号」に置き換えるのだよ。

ピーター：えーと“P”を「111」にして“E”を「110」して…結構面倒ですね。

111 110 00111 110 1000 101 111 00101 00111 00111 110 1000

になりました。なんだか、バーコードみたいですね。

博士：よろしい。そうしたら、置き換えた文字列を全部つなげて、5文字ごとに区切ってみなさい。

ピーター：はいはい、つなげて区切ると。「11111 00011 11101 00010 11110 01010 01110 01111 10100 0」
こんな感じになりました。けど、最後の「0」だけのやつはどうしますか？

博士：0を付け加えて5文字にしておいてくれ。

ピーター：えーと最後が0が1個だけだから0をあと4つ加えてやればいいんですね。できました。

11111 00011 11101 00010 11110 01010 01110 01111 10100 00000

博士：次はこの表を使うのだ。

符号	文字	符号	文字	符号	文字	符号	文字
00000	A	01000	I	10000	Q	11000	Y
00001	B	01001	J	10001	R	11001	Z
00010	C	01010	K	10010	S	11010	(空白)
00011	D	01011	L	10011	T	11011	.
00100	E	01100	M	10100	U	11100	,
00101	F	01101	N	10101	V	11101	-
00110	G	01110	O	10110	W	11110	'
00111	H	01111	P	10111	X	11111	?

ピーター：これをどう使うんですか？今度は符号から文字に置き換えるのか！

博士：そうそう。「11111」だったら「？」に、「00011」だったら「D」にとやってゆけばいいんだよ。

ピーター：これは単純ですね……えーと「?D-C' KOPUA」になりました。でも意味不明ですよ。

博士：文字数を数えてごらん。

ピーター：10文字ですよ。あっ、「PETER POTTER」は12文字だったのに2文字減っています。

博士：そう、この表を使えば文字を減らすことができるのだよ。じゃあ今度はこの文章で同じことをやっ
てごらん。

PETER PIPER PICKED A PECK OF PICKLED PEPPERS. A PECK OF PICKLED PEPPERS
PETER PIPER PICKED. IF PETER PIPER PICKED A PECK OF PICKLED PEPPERS, WHERE'S
THE PECK OF PICKLED PEPPERS PETER PIPER PICKED?

ピーター：ぎょぎょ、行が分かれています、どうするんですか？

博士：紙の都合で3行になってしまったが、改行文字のかわりに空白がある長い1行だと思っておくれ。

ピーター：はいはい。行と行の間にあるのは空白だと。しかし面倒だな……。

博士：だったらプログラムにやらせればいいではないか。

ということで、ピーターの代わりにファイルから読み込んだ文字列を博士の変換方法に従って変換し、変換後
の文字列を出力して終了するプログラムを作成してください。

入力データのファイル名はdata16.txtとします。

入力

文字列1行(すべて半角)

出力

変換後の文字列

入力例

PETER POTTER

出力例

?D-C'KOPUA

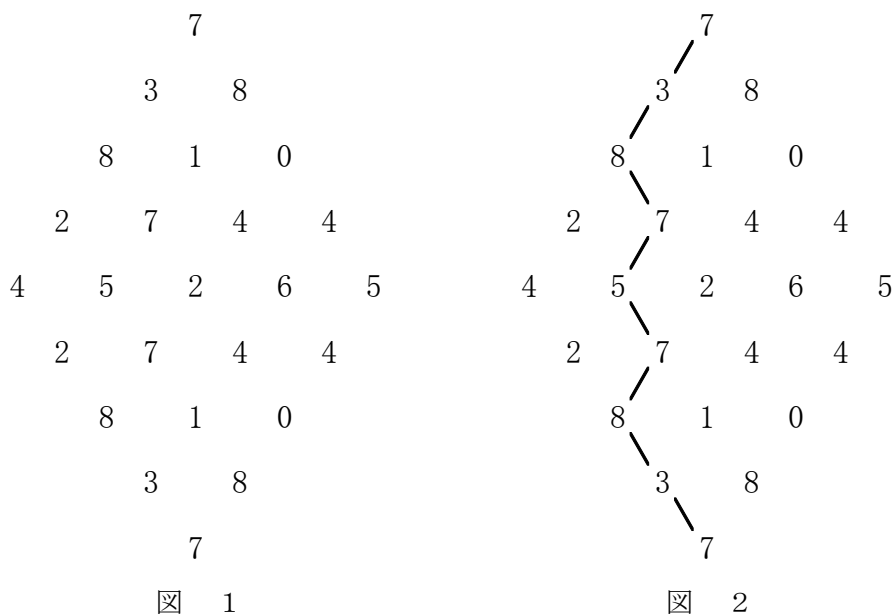


図1に例示するように整数(0以上99以下)をひしがたに並べます。このような、ひしがたを表すデータをファイルから読み込んで、一番上からスタートして一番下まで次のルールに従って進むとき、通過する整数の和の最大値を出力して終了するプログラムを作成してください。

- 各ステップで、対角線上の左下か対角線上の右下に進むことができます。

例えば図 1 の例では、図 2 に示すように、7,3,8,7,5,7,8,3,7 を選んで通ったとき、その和は最大の 55 (7+3+8+7+5+7+8+3+7=55)となります。

入力データのファイル名は data17.csv とします。

入力

図 1 のひしがたの場合、入力例のようにデータが保存されています。データの行数は、100 行未満です。

出力

ルールに従って通過する整数の和の最大値

入力例

7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
2 7 4 4
8 1 0
3 8
7

出力例

55

1 辺の長さが 10 の正方形の折り紙に半径 1 の円形のシールを n 枚貼ります。シールは重ねて貼ることができます。シールを貼る位置の座標をファイルから読み込んで、折り紙上でもっとも多くシールが重なっている場所でのシールの枚数を出力して終了するプログラムを作成してください。ただし、 n は 100 以下とします。

入力データのファイル名は data18.csv とします。

入力

折り紙の左下を原点とした x, y 座標を与えます。この x, y を円の中心としてシールを貼ることとします。円の中心が折り紙の外に出ることはありません。

x_1, y_1 (1 枚目のシールの中心座標: 全て実数)
 x_2, y_2 (2 枚目のシールの中心座標: 全て実数)
 x_3, y_3 (3 枚目のシールの中心座標: 全て実数)
:
:
 x_n, y_n (n 枚目のシールの中心座標: 全て実数)

出力

折り紙上で最も多くシールが重なっている場所でのシールの枚数(整数)

入力例

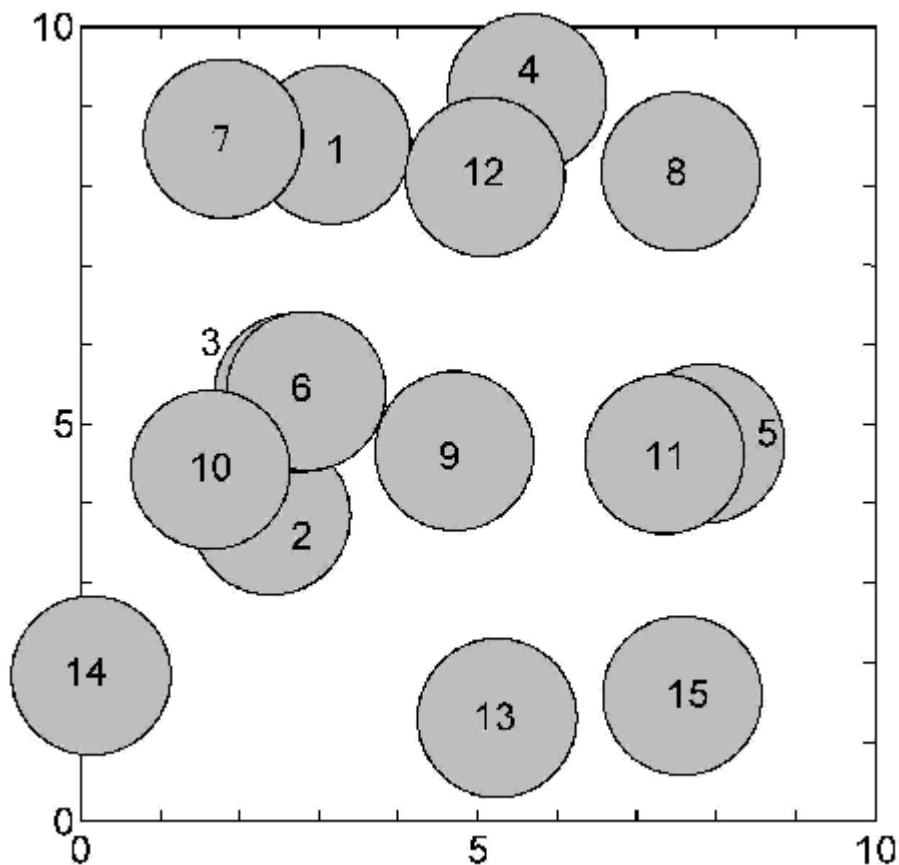
3 14979, 8 51743
2 39506, 3 84915
2 68432, 5 39095
5 61904, 9 16332
7 85653, 4 75593
2 84021, 5 41511
1 79500, 8 59211
7 55389, 8 17604
4 70665, 4 66125
1 63470, 4 42538
7 34959, 4 61981
5 09003, 8 11122
5 24373, 1 30066
0 13517, 1 83659
7 57313, 1 58150

出力例

4

参考

入力例のようにシールを貼った図です。円はシール、数字は入力例の行数を表しています。
 点(2.3,4.6)では、入力例の2行目、3行目、6行目、10行目の4枚のシールが重なっています。



6と9のおおのの中心の距離は2.01293なので、シールは重なっていません。1と12のおおのの中心の距離は1.98231なので、シールは重なっています。

2つの円が接しているとき(おおのの中心の距離が2のとき)は、重なっているものとします。

図1のような10×10のマスの「布」があり、(1, 2)のようにX座標・Y座標の値の対でマス目を示すことにします。座標値は0から始まる整数とします。例えば、図1の◎の座標は(1,2)です。

この「布」に染料を少しずつたらしつけて染物を作ります。たらず染料の滴には「大」「中」「小」の3サイズがあり、染料の滴の落ちたマス目を中心に図1のように周囲も色がつきます。図1で☆が中心、○が色のにじむ範囲です。

		X 方向									
		0	1	2	3	4	5	6	7	8	9
Y 方向	0								中		
	1							○	○	○	
	2		◎					○	☆	○	
	3							○	○	○	
	4			大							
	5			○					小		
	6		○	○	○				○		
	7	○	○	☆	○	○		○	☆	○	
	8		○	○	○				○		
	9			○							

図 1

「布」は、最初は「まっしろ」つまり、どのマス目も色の濃さを示す値が0とします。染料の滴が落ちるごとに値が1ずつ増えていきます。「小」が(1,2)、「中」が(3,2)に落ちた場合、各マス目の値は図2の左のようになります。染料がもったいないので、図2の右に示すようににじむ範囲が布の外にはみ出すような染料の落とし方はしないことになっています。また、同じ場所に複数の染料をたらすこともあります。

		X 方向									
		0	1	2	3	4	5	6	7	8	9
Y 方向	0	0	0	0	0	0	0	0	1	1	1
	1	0	1	1	1	1	0	0	0	1	0
	2	1	1	2	1	1	0	0	0	0	0
	3	0	1	1	1	1	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0
	:										

図 2

このような作業を何回か繰り返した結果、布にすばらしい模様が浮かび上がりましたが、残念なことに、うっかり作業経過を記録し忘れてしまいました。どうにもさっぱり思い出せませんが、かろうじて、落とした染料の滴数は覚えていました。あなたは、すばらしい染物を再現しなければなりません。すばらしい染物のデータをファイルから読みこんで、どこに、どのような染料をたらしただかを出力して終了するプログラムを作成してください。なお、落とした染料の滴数は12以下とします。

入力データのファイル名は data19.csv とします。

入力

最初の 1 行に落とした染料の滴数が保存されています。次の行から 10 行に各座標の色の濃さが保存されています。入力例を見てください。

出力

染料の滴の「大」を 3、「中」を 2、「小」を 1 で表し、次のように出力してください。出力行は順不同です。

X 座標,Y 座標,滴の大きさ(全て整数 カンマ区切り)

: (全て整数 カンマ区切り)

:

入力例

```
6
0,0,1,0,0,0,0,0,0,0
0,1,1,1,0,0,0,0,0,0
1,1,1,1,1,1,1,1,0,0
0,1,1,1,1,1,1,1,0,0
0,0,1,1,1,3,1,1,0,0
0,0,1,1,3,1,1,1,0,0
0,0,1,1,1,1,1,1,1,0
0,0,1,1,1,1,1,1,1,1
0,0,0,0,0,0,1,1,1,0
0,0,0,0,0,0,0,1,0,0
```

出力例

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

縦に10行、横に10列並べられた、合計100のマス目があります。いくつかのマス目には印がついています。各マス目の印の状態をファイルから読み込み、印のついていないマス目だけからなる最大の正方形の辺の長さを出力として表示するプログラムを作成してください。

入力データは1行10文字から構成され、10行が与えられます。たとえば以下のようなデータが与えられます。

```
...*. **
.....
**... **
.....*
.*.....
.....
.*.....
.....
...*. ***
.*...*...
```

入力データの一行が、一行のマス目を表現します。入力データの文字列のうち、.(ピリオド)は印のついていないマス目、*(アスタリスク)は印のついているマス目を示しています。入力データの文字列には、ピリオド、アスタリスク、改行以外の文字は含まれません。

上記の例では、下図の0で示される正方形が最大となります。

```
...*. **
.....
**... **
... 0000*
.*0000.
... 0000.
.* 0000.
... 0000.
...*. ***
.*...*...
```

よって、5と出力すれば正解になります。

なお、すべてのマス目に印がついている場合には、0を出力してください。

入力データのファイル名は data20.txt とします。

入力

マス目の状態を表現する、ピリオドとアスタリスクからなる文字列

出力

最大の正方形の辺の長さ(整数)

入力例1

```
...*. **
.....
**...*.
.....*
..*.....
.....
.*.....
.....
.....*...
..*...*
```

出力例1

5

入力例2

```
****. ****
*..*.*...
****.*...
*...*...
*... ****
.....
****. ****
*..*.*...
****.*...
*...*...
```

出力例2

3