

パソコン甲子園2004

全国高等学校パソコンコンクール

プログラミング技術部門 予選大会問題

平成 16 年 9 月 4 日 (土) 午前 10 時 ~ 正午 (時間厳守)

1 作成するプログラム

- どの問題から解答してもかまいません。
- コンソールアプリケーションを作成してください。
- 作成したプログラムはフロッピーディスクに保存して提出します。作成するプログラムのファイル名は半角英字 A+3 桁の問題番号 + 拡張子となります。ソースファイルと実行ファイルをフロッピーディスクに保存してください。
- 1つの問題に対する解答プログラムは1つとしてください。複数の解答があった場合は得点となりませんのでご注意ください。たとえば、問題 001 の解答プログラムを C と BASIC の両方で作成しても得点とはなりません。

2 データ型

- 特に断りが無い限り、問題中にある「整数」、「実数」、「文字」はそれぞれ、コンピュータで自然に扱うことができる整数値、実数値、文字列を言います。

3 入力

- 特に断りが無い限り、不正なデータは入力されませんので、不要なエラー処理を記述する必要はありません。
- 特に断りが無い限り、キーボードから入力するプログラムを作成してください。
- ファイル名が問題文中に明記されている場合はファイルをオープンして入力するプログラムを作成してください。ファイル名の拡張子が.csv となっているものは、1行に複数のデータを含み、行のデータが半角のカンマで区切られています(問題にある入力例を参考にしてください)。ファイル名の拡張子が.txt となっているものは、1行につき1つのデータしかありません。ファイルをオープンするタイプの問題では、ファイルは正しく EOF で終了していると仮定してかまいません。

4 出力

- プログラムの実行結果を画面に出力するプログラムを作成してください。問題で求められている結果のみを出力し、余計な文字や数字を出力しないようにしてください。(出力例を参考にしてください。)ただし、処理系によっては余分な空白が出力されてしまうものがありますので、それについては認めます。入力を促すメッセージプロンプトも必要ありません。

問題 001 (4点)

正の整数をファイルから読み込んで、その和を出力して終了するプログラムを作成してください。入力データはファイル c:\¥pckosien¥001.txt に保存されているものとします。

入力

正の整数データ

出力

ファイル内の整数の和

入力例

100
200
10
80
5
2
3

出力例

400

問題 002 (4点)

2004年1月1日から12月31日までの日付を入力し、その日の曜日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、2004年1月1日は木曜日です。また、今年はうるう年ですので2月29日があることに注意してください。

入力

月(整数)

日(整数)

出力

曜日

入力例 1

1

1

出力例 1

木曜日

入力例 2

2

29

出力例 2

日曜日

問題 003 (4点)

1~100の整数データをファイルから読みこんで、データの最頻値^{*1}を出力して終了するプログラムを作成してください。

入力データは、ファイル c:\¥pckosien¥003.txt に保存されているものとします。データ数は100件以内です。

入力

整数データ(100件以内)

出力

整数データの最頻値(複数ある場合は、改行してすべて表示)

入力例

5
6
5
8
7
5
9
7
3
4

出力例

5

^{*1} 最頻値とは、最も頻度の高い(出現回数が多い)値のことです。入力例を見てください。「5」という数値が3つ、「7」は2つ、他は1つです。このとき、もっとも個数の多い「5」が最頻値となります(出力例)。

問題 004 (4点)

英語の文章に出現する単語の中で、文字数が最も多い単語を出力して終了するプログラムを作成してください。入力データには英文字、スペースのみが含まれているものとします。単語とは、スペースで区切られた連続する英文字の列です。文章の文字数は 80 文字以下です。一つの単語の文字数は 32 文字以下です。最長の文字数を持つ単語は文中にただ一つだけ存在します。

入力

英語の文章を表現する文字列 (半角の英文字、スペース)

出力

最も文字数が多い単語 1 個

入力例

```
Good medicine tastes bitter
```

出力例

```
medicine
```


問題 005 (7点)

0 から 9 の数字から異なる n 個の数を取り出して合計が s となる組み合わせの数を出力して終了するプログラムを作成してください。

n 個の数はおののおの 0 から 9 までとし、1 つの組み合わせに同じ数字は使えません。

たとえば、 n が 3 で s が 6 のとき、3 個の数字の合計が 6 になる組み合わせは、

$$1 + 2 + 3 = 6$$

$$0 + 1 + 5 = 6$$

$$0 + 2 + 4 = 6$$

の 3 通りとなります (入力例 1 と出力例 1 をご覧ください)。

入力

n (整数)

s (整数)

出力

n 個の整数の和が s になる組み合わせの数 (整数)

入力例 1

3

6

出力例 1

3

入力例 2

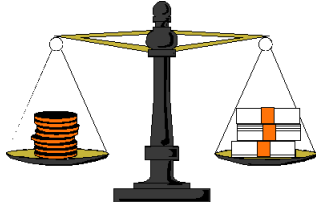
3

1

出力例 2

0

問題 006 (7点)



祖母が天秤を使っています。天秤は、二つの皿の両方に同じ目方のものを載せると釣り合い、そうでない場合には、重い方に傾きます。10個の分銅の重さは、軽い順に 1g, 2g, 4g, 8g, 16g, 32g, 64g, 128g, 256g, 512g です。

祖母は、「1kg くらいまでグラム単位で量れるのよ。」と言います。「じゃあ、試しに、ここにあるジュースの重さを量ってよ」と言ってみると、祖母は左の皿にジュースを、右の皿に 8g と 64g と 128g の分銅を載せて釣合わせてから、「分銅の目方の合計は 200g だから、ジュースの目方は 200g ね。どう、正しいでしょう?」と答えました。

左の皿に載せる品物の重さを与えるので、天秤で与えられた重みの品物と釣り合わせる時に、右の皿に載せる分銅を軽い順に空白で区切って出力して終了するプログラムを作成して下さい。ただし、量るべき品物の重さは、すべての分銅の重さの合計 (=1023g) より小さいものとします。

ヒント： 分銅の重さは $2^n (n = 0, 1, \dots, 9)$ g です。

入力

左の皿に載せる品物の重さ (正の整数)

出力

右の皿に載せる分銅 (昇順) 空白区切り

入力例 1

5

出力例 1

1 4

入力例 2

7

出力例 2

1 2 4

入力例 3

127

出力例 3

1 2 4 8 16 32 64

問題 007 (7点)

機械に辺・対角線の長さのデータを入力し、プラスチック板の型抜きをしている工場があります。この工場では、サイズはいろいろですが、平行四辺形の型のみを切り出しています。あなたは、切り出される平行四辺形のうち、長方形とひし形の製造個数を数えるように上司から命じられました。

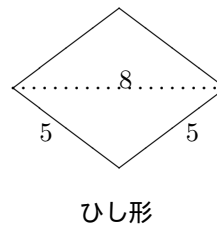
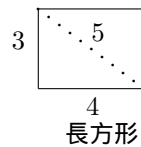
「機械に入力するデータ」をファイルから読み込んで、長方形とひし形の製造個数を出力して終了するプログラムを作成してください。

入力データはファイル `c:\¥pckosien¥007.csv` に保存されているものとします。

ヒント

長方形 全ての角の角度が等しい四角形

ひし形 全ての辺の長さが等しい四角形



入力

「機械に入力するデータ」は、平行四辺形の隣り合う2辺の長さとおののの長さが記録されています。おののの長さは全て整数で与えられます。入力例を参考にしてください。

辺1の長さ, 辺2の長さ, 対角線の長さ (1件目のデータ: 全て整数)
:
(2件目のデータ: 全て整数)
:
(3件目のデータ: 全て整数)
:
:

出力

長方形の製造個数
ひし形の製造個数

入力例

3,4,5

5,5,8

4,4,4

5,4,3

出力例

1

2

問題 008 (7点)

図のように二股に分かれている容器があります。1 から 10 までの番号が付けた 10 個の玉を容器の開口部 A から落とし、左の筒 B か右の筒 C に玉を入れます。板 D は支点 E を中心に左右に回転できるので、板 D を動かすことで筒 B と筒 C のどちらに入れるか決めることができます。

開口部 A から落とす玉の並びを与えます。それらを順番に筒 B 又は筒 C に入れていきます。このとき、筒 B と筒 C のおのおのが両方とも番号の小さい玉の上に大きい玉を並べられる場合は YES、並べられない場合は NO と半角大文字で出力して終了するプログラムを作成してください。

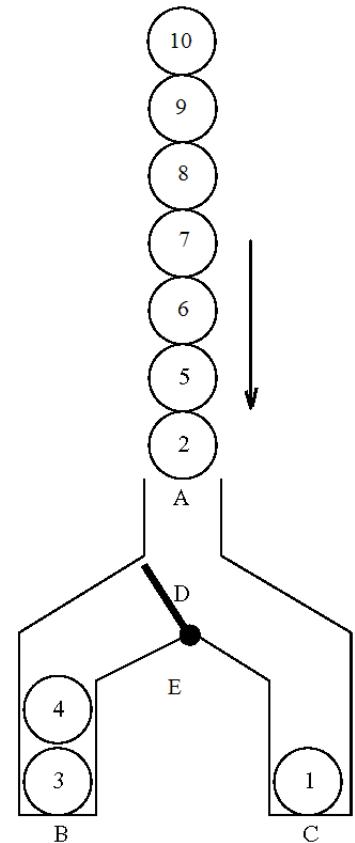
ただし、容器の中で玉の順序を入れ替えることはできないものとします。また、続けて同じ筒に入れることができるものとし、筒 B, C ともに 10 個の玉がすべて入るだけの余裕があるとします。

入力

- 1 番目の玉の番号 (整数)
- 2 番目の玉の番号 (整数)
- 3 番目の玉の番号 (整数)
- 4 番目の玉の番号 (整数)
- 5 番目の玉の番号 (整数)
- 6 番目の玉の番号 (整数)
- 7 番目の玉の番号 (整数)
- 8 番目の玉の番号 (整数)
- 9 番目の玉の番号 (整数)
- 10 番目の玉の番号 (整数)

出力

YES または NO



入力例 1

3
1
4
2
5
6
7
8
9
10

出力例 1

YES

入力例 2

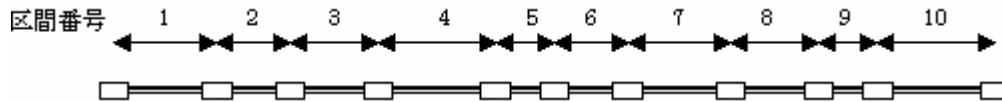
10
9
8
7
6
5
4
3
2
1

出力例 2

NO

問題 009 (7点)

複線（上りと下りが別の線路になっていてどこでもすれ違える）の鉄道路線があります。この路線には終端駅を含めて 11 の駅があり、それぞれの駅と駅の間は図で示す区間番号で呼ばれています。



この路線の両方の終端駅から列車が同時に出発して、途中で停まらずに走ります。各区間の長さとして 2 本の列車の速度をファイルから読み込んで、それぞれの場合について列車がすれ違う区間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、ちょうど駅のところですれ違う場合は、両側の区間番号のうち小さいほうの数字を出力します。また、列車の長さ、駅の長さは無視できるものとします。

入力データはファイル `c:\¥pckosien¥009.csv` に保存されているものとします。

入力

区間 1 の長さを l_1 km、区間 2 の長さを l_2 km、...、区間 10 の長さを l_{10} km、区間 1 側の終端駅を出発した列車の速度を v_1 km/h、区間 10 側の終端駅を出発した列車の速度を v_2 km/h としたとき、入力データは以下ようになります。

$l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, l_{10}, v_1, v_2$ (1 件目のデータ：全て整数)
 ⋮ (2 件目のデータ：全て整数)
 ⋮ (3 件目のデータ：全て整数)
 ⋮ ⋮

出力

列車がすれちがう区間の番号 (1 件目のデータに対する出力)
 ⋮ (2 件目のデータに対する出力)
 ⋮ (3 件目のデータに対する出力)
 ⋮ ⋮

入力例

1,1,1,1,1,1,1,1,1,1,40,60
 1,1,1,1,1,3,3,3,3,3,50,50
 10,10,10,10,10,10,10,10,10,10,50,49

出力例

4
7
6

問題 010 (7点)

平面上の異なる4点 $A(x_A, y_A), B(x_B, y_B), C(x_C, y_C), D(x_D, y_D)$ の座標をファイルから読み込んで、それぞれについて直線 AB と CD が平行である場合には YES、平行でない場合には NO と半角大文字で出力して終了するプログラムを作成してください。なお、 $x_A, y_A, x_B, y_B, x_C, y_C, x_D, y_D$ は、それぞれ -100 以上 100 以下の実数とします。

入力データはファイル `c:\¥pckosien¥010.csv` に保存されているものとします。

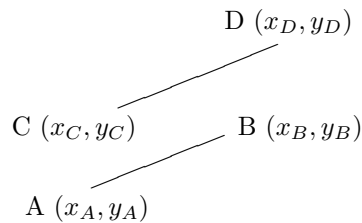


図 1

入力

$x_A, y_A, x_B, y_B, x_C, y_C, x_D, y_D$ (1 件目のデータ: 全て実数)
⋮ (2 件目のデータ: 全て実数)
⋮ (3 件目のデータ: 全て実数)
⋮ ⋮

出力

YES または NO (1 件目のデータに対する出力)
⋮ (2 件目のデータに対する出力)
⋮ (3 件目のデータに対する出力)
⋮ ⋮

入力例

```
0.0,0.0,1.0,1.0,1.0,0.0,2.0,1.0
0.0,0.0,2.0,0.0,-1.0,2.0,2.0,2.0
10.0,6.0,3.4,5.2,6.8,9.5,4.3,2.1
3.0,2.0,9.0,6.0,13.0,5.0,7.0,9.0
2.0,2.0,2.0,9.0,3.0,5.0,3.0,-8.0
```

出力例

YES

YES

NO

NO

YES

入力例に対応するグラフ（参考）

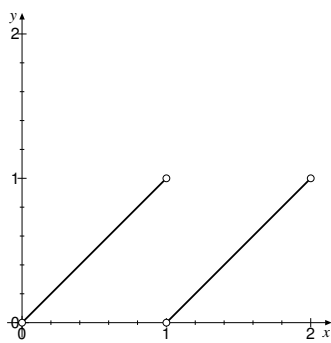


図 2（入力例の 1 行目）

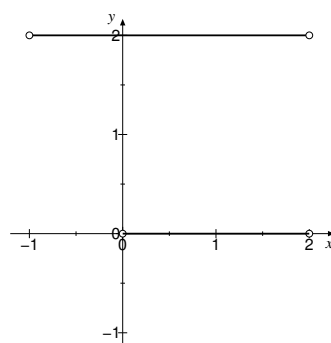


図 3（入力例の 2 行目）

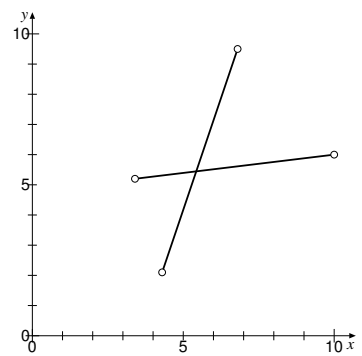


図 4（入力例の 3 行目）

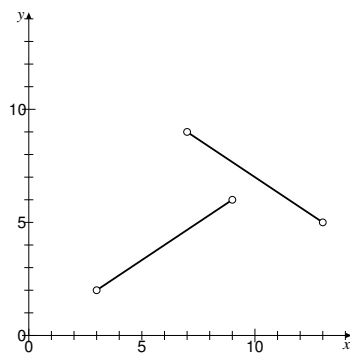


図 5（入力例の 4 行目）

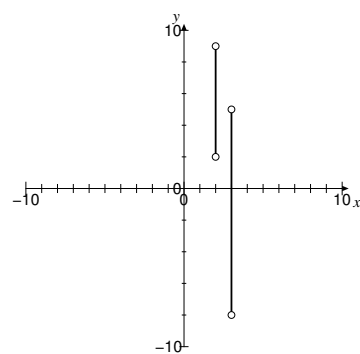


図 6（入力例の 5 行目）

問題 011 (7点)

平面上の異なる4点 $A(x_A, y_A), B(x_B, y_B), C(x_C, y_C), D(x_D, y_D)$ の座標をファイルから読み込んで、それぞれについて、それら4点を頂点とした四角形 $ABCD$ に凹みがなければ YES、凹みがあれば NO と半角大文字で出力して終了するプログラムを作成してください。なお、 $x_A, y_A, x_B, y_B, x_C, y_C, x_D, y_D$ は、それぞれ -100 以上 100 以下の実数とします。

入力データはファイル `c:\¥pckosien¥011.csv` に保存されているものとします。

凹みのある四角形とは図1のような四角形です。

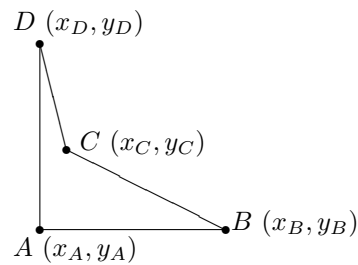


図1

なお、1直線上に3つ以上点が並ぶことはないものとします。また、入力順に点を結んでいけば、四角形になる順番に点の座標が入力されるものとします。(つまり、図2のような形になる順番はありません。)

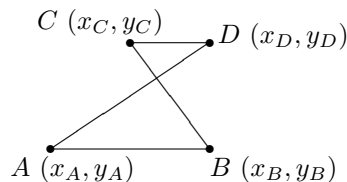


図2

入力

$x_A, y_A, x_B, y_B, x_C, y_C, x_D, y_D$ (1件目のデータ: 全て実数)
⋮ (2件目のデータ: 全て実数)
⋮ (3件目のデータ: 全て実数)
⋮ ⋮

出力

YES または NO (1件目のデータに対する出力)
⋮ (2件目のデータに対する出力)
⋮ (3件目のデータに対する出力)
⋮ ⋮

入力例

0.0,0.0,1.0,0.0,1.0,1.0,0.0,1.0

0.0,0.0,3.0,0.0,1.0,1.0,1.0,3.0

出力例

YES

NO

入力例に対応するグラフ（参考）

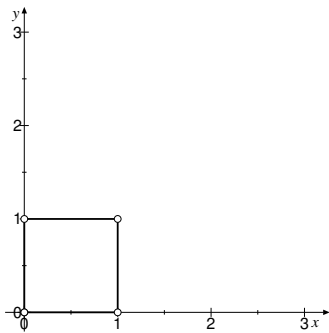


図 3（入力例の 1 行目）

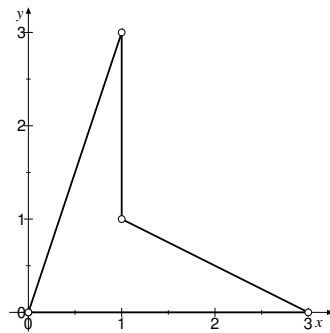


図 4（入力例の 2 行目）

問題 012 (7点)

縦 8、横 8 のマスからなる図 1 のような平面があります。

図 1

その平面上に、下の A から G の図形のどれかが一つだけ置かれています。

A

B

C

D

E

F

G

たとえば、図 2 の例では

```
00000000
00000000
01100000
00110000
00000000
00000000
00000000
00000000
```

図 2 (対応する数字の並び)

E の図形が置かれています。平面の中で図形が占めているマスを 1、占めていないマスを 0 で表現した数字の列をファイルから読み込んで、置かれている図形の種類 (A ~ G) 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、ひとつの平面に置かれている図形は必ず 1 つで、複数の図形が置かれていることはありません。また、A ~ G で表される図形以外のものが置かれていることはありません。

入力データは c:\¥pckosien¥012.txt に保存されているものとします。

入力

平面の中で図形が占めているマスを 1、占めていないマスを 0 で表現した 8 文字からなる文字列が 8 行でひとつの平面となります。図 2 の場合に対応する数字の並びは図 2 の右にあるとおりです。平面と平面の区切り

を空行で表します。入力例を見てください。

出力

各平面に与えられた図形の種類 (A ~ G のいずれか) を半角大文字で 1 行にひとつずつ出力してください。
出力例を見てください。

入力例

```
00000000
00000000
01100000
00110000
00000000
00000000
00000000
00000000
```

```
00011110
00000000
00000000
00000000
00000000
00000000
00000000
00000000
```

```
00000000
00000000
00110000
00110000
00000000
00000000
00000000
00000000
```

出力例

```
E
C
A
```

問題 013 (7点)

縦線が5本のあみだくじがあります。このあみだくじは、

- 横線は真横に引きます。斜めに引くことはありません。
- 横線は必ず隣り合った縦線同士をつなぎます。つまり、横線が縦線をまたぐことはありません。
- どの縦線についても同じ点から左右同時に横線が出ることはありません。つまり、横線が縦線を横切ることはありません。
- 当りはひとつだけです。

図1にあみだくじの例を示します。上側の数字は縦線の番号(左から1, 2, 3, 4, 5)を表します。 が当たりです。

1	2	3	4	5
				1010
				1001
				0100
				1001
				0010
				1000
				0100
				0101
				1010

図1 (対応する数字の並び)

あみだくじの当りの場所、横線の有無をファイルから読み込んで、当りになる縦線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。

入力データはファイル c:\¥pckosien¥013.txt に保存されているものとします。

入力

1行目に、当りの場所(図1でいう)が左から数えて何番目かが記録されています。

2行目以降は、図1に対応する数字の並びのように、あみだくじの上から順に、各縦線の間横線があるときを1、ないときを0として、4つの数字が並んでいます。

入力例を参考にしてください。

出力

スタート地点で左から何番目の縦線を選ぶと当りを引くことができるかを出力します。

入力例

図 1 のときの入力ファイルの内容は以下のようになっています。

```
3
1010
1001
0100
1001
0010
1000
0100
0101
1010
```

出力例

```
5
```

問題 014 (7点)

上から見ると図1のような形の格子状の広場があります。この格子の各辺に「壁」があるかないかを0と1の並びで示しています。点Aに立って壁に右手をつき、壁に右手をついたまま、矢印の方向に歩き続けて再び点Aに戻ってくるまでの経路を出力して終了するプログラムを作成してください。

入力データはファイル c:\¥pckosien¥014.txt に保存されているものとします。

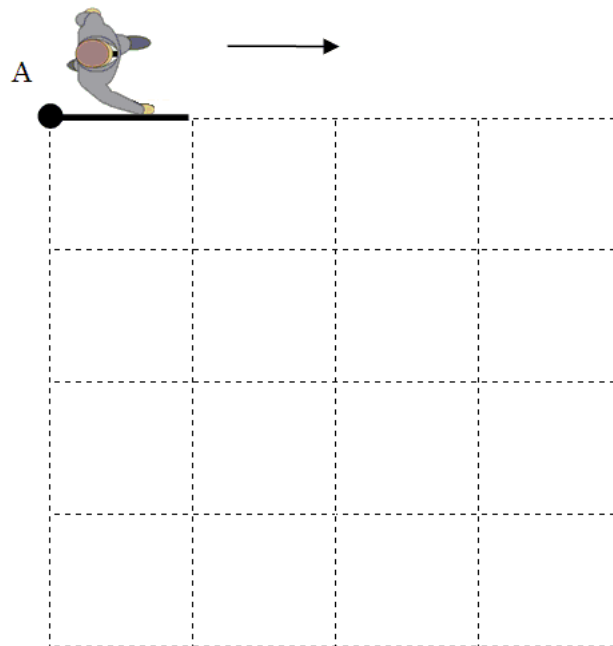


図1

入力

図2に示すように壁がある場合を1、ない場合を0として

- 1行目は一番上の横線の壁の有無を左から0と1で表した文字列
- 2行目はその下の縦線の壁の有無を左から0と1で表した文字列
- 3行目は上から2本目の横線の壁の有無を左から0と1で表した文字列
- ⋮
- 9行目は一番下の横線の壁の有無を左から0と1で表した文字列

ただし、図1の太線に示すように、点Aから右に1区画分は必ず壁があるものとします。すなわち、1行目の1文字目は常に1です。

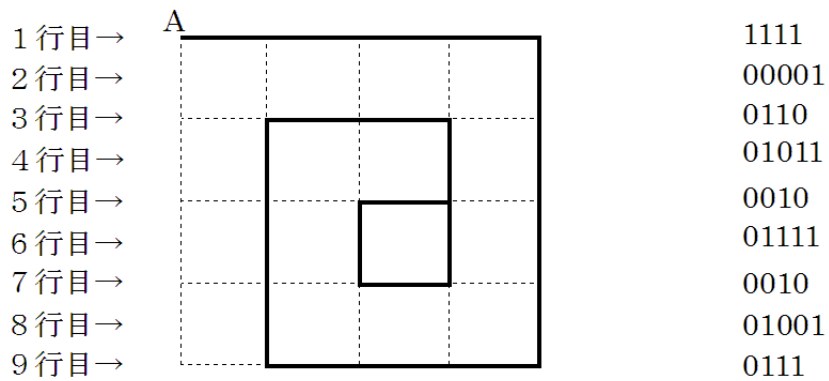


図 2 (壁があるところを太線で表しています) (対応する数字の並び)

出力

「図の左方向に一区画分進む」を L で
「図の右方向に一区画分進む」を R で
「図の上方向に一区画分進む」を U で
「図の下方向に一区画分進む」を D で
表わし、進む順に L R U D を 1 行に並べて出力します。文字は全て半角大文字です。

入力例

図 2 のときのファイルの内容は以下のようになっています。

```
1111
00001
0110
01011
0010
01111
0010
01001
0111
```

出力例

```
RRRRDDDDLLLJUURRDDLURULLDDRRRUUUULLLL
```

問題 015 (7 点)

ポーカーの手札データをファイルから読み込んで、それぞれについてその役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、この問題では、以下のルールに従います。

- ポーカーはトランプ 5 枚で行う競技です。
- 同じ数字のカードは 5 枚以上ありません。
- ジョーカーは無いものとします。
- 以下のポーカーの役だけを考えるものとします。(下にいくにつれて役が高くなります。)
 1. 役なし (以下に挙げるどれにも当てはまらない)
 2. ワンペア (2 枚の同じ数字のカードが 1 組ある)
 3. ツーペア (2 枚の同じ数字のカードが 2 組ある)
 4. スリーカード (3 枚の同じ数字のカードが 1 組ある)
 5. ストレート (5 枚のカードの数字が連続している)

A を含むストレートの場合、A で終わる並びもストレートとします。つまり、A を含むストレートは、A 2 3 4 5 と 10 J Q K A の 2 種類です。J Q K A 2 などのように、A をまたぐ並びはストレートではありません。(この場合、役なし になります)
 6. フルハウス (3 枚の同じ数字のカードが 1 組と、残りの 2 枚が同じ数字のカード)
 7. フォーカード (4 枚の同じ数字のカードが 1 組ある)

入力データはファイル c:\¥pckosien¥015.csv に保存されているものとします。

入力

トランプの J(ジャック) を 11、Q(クイーン) を 12、K(キング) を 13、A (エース) を 1 のそれぞれの数字で表すこととします。入力例を見てください。

手札 1, 手札 2, 手札 3, 手札 4, 手札 5 (1 件目のデータ : 正の整数)
 ⋮ (2 件目のデータ : 正の整数)
 ⋮ (3 件目のデータ : 正の整数)
 ⋮ ⋮

出力

手札によってできるもっとも高い役をひとつ出力するようにしてください。たとえば、3 3 2 3 3 という手札であったときは、 ツーペア ではなく フォーカードです。出力例を見てください。

役名 (1 件目のデータに対する出力)
∴ (2 件目のデータに対する出力)
∴ (3 件目のデータに対する出力)
∴ ∴

入力例

1, 2, 3, 4, 1
2, 3, 2, 3, 12
12, 13, 11, 12, 12
7, 6, 7, 6, 7
3, 3, 2, 3, 3
6, 7, 8, 9, 10
11, 12, 10, 1, 13
11, 12, 13, 1, 2

出力例

ワンペア
ツーペア
スリーカード
フルハウス
フォーカード
ストレート
ストレート
役なし

問題 016 (7点)

古代ローマでは数を数えることは難しい仕事でした。アラビア数字の 0,1,2,3,..., 9 はまだ流布していませんでした。その代わり次のような記号が使われていました。

アラビア数字	ローマ数字	アラビア数字	ローマ数字	アラビア数字	ローマ数字
1	I	11	XI	30	XXX
2	II	12	XII	40	XL
3	III	13	XIII	50	L
4	IV	14	XIV	60	LX
5	V	15	XV	70	LXX
6	VI	16	XVI	80	LXXX
7	VII	17	XVII	90	XC
8	VIII	18	XVIII	100	C
9	IX	19	XIX	500	D
10	X	20	XX	1000	M

I は 1, V は 5, X は 10, L は 50, C は 100, D は 500, M は 1000, 他の例は上の表を見てください。小さい数が大きい数に続いている、つまり右側にあるときは足し算をします。小さい数が大きい数の前に、つまり左にあるときは、大きい数から小さい数を引きます。大きい数字の前にあって引き算を表す小さな数字は一回の引き算あたりひとつしかありません。

ローマ数字をアラビア数字(通常の数値)の表記(10進表示)に変換して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、与えられるローマ数字の文字列の長さはおのこの 100 以下です。また、与えられるローマ数字は上記のルールにのみ従っています*2。

入力

ローマ数字(半角大文字の I, V, X, L, C, D, M で表される連続した文字列)

*2 実際のローマ数字の表記にはもっと細かいルールがありますが、ここでは考慮する必要はありません。たとえば、実際のローマ数字では I は V か X から、X は L か C から、C は D か M からしか引き算しませんし、同じローマ数字は 4 つ以上(または 5 つ以上) 足し並べることはありません。

出力

アラビア数字 (整数)

入力例 1

IV

出力例 1

4

入力例 2

CCCCLXXXVIII

出力例 2

499

入力例 3

CDXCIX

出力例 3

499

問題 017 (25 点)

簡単な暗号法の一つに、アフィン暗号というものがあります。

まず、アルファベット $a \sim z$ を $a=0, b=1, c=2, \dots, x=23, y=24, z=25$ と $0 \sim 25$ の数字に置き換えます。

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

そして、以下の式で、原文のアルファベットを置換します。

$$e(\gamma) = f_{26}(\alpha \cdot \gamma + \beta)$$

ただし、 $f_{26}(x)$ は x を 26 で割ったあまりを表す関数とします。

たとえば、 $\alpha = 3, \beta = 2$ のとき、アルファベットの 'a' (=0) は、 $e(0) = f_{26}(3 \times 0 + 2) = f_{26}(2) = 2$ で 'c' に、アルファベットの 'n' (=13) は $e(13) = f_{26}(3 \times 13 + 2) = f_{26}(41) = 15$ で 'p' に置換されます。

このとき、 γ に対する $e(\gamma)$ が必ず 1 対 1 で対応付けられるように、 α と β は慎重に選ばれているものとします*3。 $\alpha = 4, \beta = 7$ のときのように、 $e(a) = f_{26}(4 \times 0 + 7) = f_{26}(7) = 7$, $e(n) = f_{26}(4 \times 13 + 7) = f_{26}(59) = 7$ と、'a' も 'n' も同じ 'h' に置換されるようなことはありません。

また、アルファベット以外の文字は置換されません。

暗号化された文字列を元の文章に復号したものを出力して終了するプログラムを作成してください。

元の文章には、キーワードとして

that

this

のいずれかが必ず含まれていることとします。

入力

半角の英小文字と空白からなる 256 文字以内の暗号化された文章

出力

復号した元の文章

*3 α と 26 が互いに素 (最大公約数が 1) であることが条件です。

入力例

y eazqyp pnop pngtg ye obmpngt xmybp mr lygw

出力例 ($\alpha = 11, \beta = 14$ となる)

i submit that there is another point of view

問題 018 (25 点)

与えられた 4 つの 1~9 の整数を使って、答えが 10 になる式をつくります。

4 つの整数を入力したとき、下記の条件に従い、答えが 10 になる式を出力して終了するプログラムを作成^{*4}してください。また、答が複数ある時は、最初に見つかった答だけを出力するものとします。答がない時は、0 と出力してください。

- 演算子として、加算 (+)、減算 (-)、乗算 (*) だけを使います。除算 (/) は使いません。
- 数を 4 つとも使わなければいけません。
- 4 つの数の順番は自由に入れ換えてかまいません。
- カッコを使ってもかまいません。

入力

1~9 の整数

1~9 の整数

1~9 の整数

1~9 の整数

出力

与えられた 4 つの整数と上記の演算記号およびカッコを組み合わせると値が 10 となる式または 0

入力例 1

8

7

9

9

出力例 1

$((9 * (9 - 7)) - 8)$

または

$((9 - 7) * 9) - 8)$

^{*4} 出力例のように、(左の項 演算子 右の項) と、すべての演算をカッコでくくって出力するように考えるとよいでしょう。

入力例 2

4
4
4
4

出力例 2

0

入力例 3

5
5
7
5

出力例 3

$((7 * 5) - (5 * 5))$

または

$((5 * 7) - (5 * 5))$

問題 019 (25 点)



宝物がたくさん収蔵されている博物館に、泥棒が大きな風呂敷を一つだけ持って忍び込みました。盗み出したいものはたくさんありますが、風呂敷が耐えられる重さが限られており、これを超えると風呂敷が破れてしまいます。そこで泥棒は、用意した風呂敷を破らず且つ最も価値が高くなるようなお宝の組み合わせを考えなくてはなりません。

風呂敷が耐えられる重さ W 、および博物館にある個々のお宝の価値と重さをファイルから読み込んで、重さの総和が W を超えない範囲で価値の総和が最大になるときの、お宝の価値総和と重さの総和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、価値の総和が最大になる組み合わせが複数あるときは、重さの総和が小さいものを出力することとします。また、博物館にあるお宝の総数は 100 以下とし、お宝の価値は 10000 以下とします。

入力データはファイル `c:\¥pckosien¥019.csv` に格納されているものとします。

入力

風呂敷の耐えられる重さ W (1000 以下)
お宝 1 の価値 (整数), お宝 1 の重さ ($<W$:整数)
お宝 2 の価値 (整数), お宝 2 の重さ ($<W$:整数)
⋮

出力

風呂敷に入れたお宝の価値総和
そのときのお宝の重さの総和

入力例

50
60,10
100,20
120,30
210,45
10,4

出力例

220

49

問題 020 (25 点)

1~9 の数字を 14 個組み合わせるパズルがあります。与えられた 13 個の数字にもうひとつ数字を付け加えて完成させます。

13 個の数字からなる文字列を、ファイルから読み込んで、それぞれについて、パズルを完成することができる数字を昇順に全て出力して終了するプログラムを作成してください。なお、1~9 のどの数字を付け加えてもパズルを完成させることができないときは 0 を出力してください。

入力データはファイル `c:\¥pckosien¥020.txt` に保存されているものとします。

パズルの完成条件

- 同じ数字を 2 つ組み合わせたものが必ずひとつ必要です。
- 残りの 12 個の数字は、3 個の数字の組み合わせ 4 つです。
3 個の数字の組み合わせ方は、同じ数字を 3 つ組み合わせたものか、または 3 つの連続する数字を組み合わせたものです。ただし、9 1 2 のような並びは連続する数字とは認められません。
- 同じ数字は 4 回まで使えます。

ヒント 3456666777999 の例では、

「2」があれば、 234 567 666 77 999

「3」があれば、 33 456 666 777 999

「5」があれば、 345 567 666 77 999

「8」があれば、 345 666 678 77 999

というふうに、2 3 5 8 のいずれかの数字が付け加えられるとパズルは完成します。「6」でも整いますが、5 回目の使用になるので、この例では使えないことに注意してください。

入力

13 個の数字 (1 件目のデータ)

⋮ (2 件目のデータ)

⋮ (3 件目のデータ)

⋮ ⋮

出力

パズルを完成させることができる数字 (1 件目のデータに対する出力: 昇順 空白区切り)
:
(2 件目のデータに対する出力: 昇順 空白区切り)
:
(3 件目のデータに対する出力: 昇順 空白区切り)
:
:

入力例

3649596966777
6358665788577
9118992346175
9643871425498
7755542764533
1133557799246

出力例

2 3 5 8
3 4
1 2 3 4 5 6 7 8 9
7 8 9
1 2 3 4 6 7 8
0