

リクルートコミュニケーションズ エンジニア採用課題

- 6問中 最低2問以上を解答してください。
- 現在のメイン開発言語がPerl、Ruby、Python、Java、Scala、Node.jsとなるため、いずれかの開発言語でコーディングをお願いします。
- 開発環境は問いませんが最終的にLinux上で動作するようにコーディングをお願いします。
- 処理速度も評価対象となります。
- 提出されたコードは原則全て拝見させていただきますが、コード量が大きくなりすぎる場合は評価しきれないことがありますのでコンパクトなコードを心がけてください。
- また、以下のケースは採点不能となりますのでご注意ください。
 - コンパイルできない
 - バージョン、ライブラリがマイナーすぎて入手が困難
 - ネット上資源の流用や質問サイトの使用
- 提出いただいた課題、および回答はお返しいたしませんのでご了承ください。

問題 1

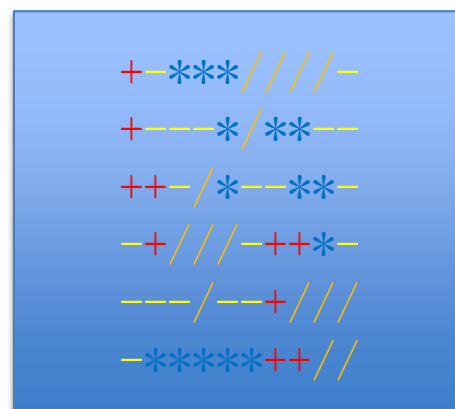
下記例の左図のような入力ファイルに対し、隣接するブロック(同じ文字のマスが同じブロック)が同じ色にならないように四色に塗り分けるプログラムを作成してください。

例



```
023337777a
02223788aa
002436688a
104446998a
1114669bbb
1555599bb
```

入力例



```
+---***/////
+---*/*-
++-/*-*-
-+///-++*-
---/-++///
-*****++//
```

出力例

- 入力ファイルのサイズは最大1000 × 1000マス
 - 各ブロック内のマスは上下左右いずれかの面で同ブロック内の他のマスとつながっている
 - 同じブロックのマスには同じ記号が割り振られ、記号は0-9a-zA-Zで最大62ブロック
 - 出力ファイルは入力ファイルと同じサイズ
 - +-*//の4つの記号を4色とし、各マスに割り振ってください
 - 同じブロック内のマスは同じ色
 - 上下左右いずれかの面で隣接するブロック同士は同じ色にはできない
- 解答が複数パターンある場合は任意の1パターンを出力してください。
解答が存在しない場合は「解答不可」の旨を出力してください。

問題 2

標準入力から与えられた数式を評価し、結果を標準出力に出力する簡易計算機プログラムを作成してください。

入力として、以下のような形式のテキストが標準入力から与えられます。

$x+4-(y-30)*2$

$x=1,y=0$

$x=5$

$y=-2$

$z=10$

1行目は数式です。

次に空行が1行あり、3行目以降の各行に状態(変数の値)が与えられます。

数式を各状態で評価した結果を、標準出力に改行とともに出力してください。

この例の場合は、

65

$-2y+69$

$x+68$

$x-2y+64$

と出力します。

この例の場合、2行目から下の状態には式に現れた全ての変数が現れないため、上記のように変数をそのまま出力します。

q2ディレクトリに、input0.txt から input9.txt まで他の入力例がありますので参考にしてください。

対応する出力例は output0.txt から output9.txt です。

●入力と数式

3行目以降の状態(変数の値)は、最大で1000行まで現れます。

数式の長さは最大で1000文字です。括弧の入れ子の数に制限はありません。

数式に現れる要素は以下のとおりです。

- 演算子は、`+` と `-` と `*` で、それぞれ加算、減算、乗算を表す
- 括弧は `()` が現れ、入れ子構造もととり得る。
- 定数は整数のみ(小数は現れない。単項のマイナスは現れる。`-a+b` や `-3*c` など)
- 現れる変数は1文字のみ `a` ~ `z` のいずれか

●演算と出力

以下のようなルールで数式を評価して出力してください。

- 演算のルールは一般的な数学のルールに則る
 - 演算子の優先順位は `()` > `*` > `+` or `-`
 - 評価は左結合
- 数式に現れない変数は無視する
- 空白は現れない
- 回答に未知の変数が存在しない場合は、単にその値を出力する (例) 4, -10, 0 など
- 未知の変数が存在する場合は、それらを多項式として出力する。
- 各項を次数が高い順に出力し次数が同じ時はアルファベット順に出力する。2乗、3乗などは後ろに数字を置いて表現する (例) `2*a*a*b-4*a*b*b+b*c+a+40` は `2a2b-4ab2+bc+a+40` と表現

●制限事項

- eval などに代表される、言語処理系に式を評価させる機構は利用できない
- 言語の標準機能のみで実装する。文字列処理、正規表現、標準入出力、コンテナクラスなどがライブラリとして提供されている言語の場合は、それらのライブラリを使用しても良い
- パーサジェネレータやパーサコンビネータ、構文解析器系のライブラリは、標準ライブラリだとしても使用できない。数字を数値に変換するものなどは使用可能

問題 3

以下のルールに基づいた暗号文を解読するプログラムを作成してください。

- 暗号元の文章(以下、平文)のアルファベットをa -> d, e -> fのように1対1対応で入れ替える
- 大文字は小文字と同様に変換する(a -> d ならば A -> D)
- アルファベット以外の文字(数字や記号、空白等)は変換しない
- アルファベット入れ替えルールは暗号化時に毎回変更される

なお、平文内の単語を網羅した辞書を利用することが可能です。

辞書内の単語は平文の単語を過不足無く含んでいるものとします。

暗号文、辞書、平文の各ファイルについては以下のサンプルを参考にしてください。

- 暗号文(q3/question.txt)
- 辞書(q3/dict.txt)
- 平文(q3/answer.txt)

作成するプログラムでは暗号文ファイル(と必要であれば辞書ファイル)を引数とし標準出力に平文を出力してください。

また、復号できない場合はその旨を結果として出力してください。

問題 4

大きな画像(512px × 512px)が1枚と小さな画像(64px × 64px)が1000枚あります。

- 画像はグレースケールで各ピクセルの値は0-255です。
- 小さな画像には000-999まで番号が振られています。

大きな画像から小さな画像と同じ部分(完全一致)を探すプログラムを作成してください。

発見した全ての画像について、その左上の座標(大きい画像の左上が0,0)と画像の番号を以下のように列挙してください。(順不同)

10,20,5 ※座標(10,20)で5番の画像が完全一致

50,100,97 ※座標(50,100)で97番の画像が完全一致

存在しない場合は、「存在しません」と出力してください。

(プログラム入力引数)

1. 大きな画像のパス(拡張子はpng)
2. 小さな画像のフォルダパス(画像ファイルの拡張子はpng。ファイル名は「番号.png」)

実行例 : プログラム名 img/large.png img/small

サンプルとしてq4/large.png、q4/smallを用意しました。

このサンプルの場合、

100,100,333

300,200,666

200,350,999

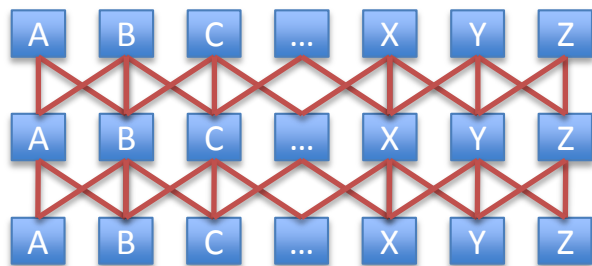
が正解になります。

※画像の読み込み以外でopenCVなどのライブラリは使用しないでください。

※採点時は、今回添付していない極端な画像も含むサンプルを用い、その中の最も遅い処理速度で評価します。

問題 5

以下のような26ノード(A..Z) x 3層のネットワークがあります。



1層目は A..Z から選択し、2層目以降はつながっている経路のみ通れます。

OK例: A->A->A, C->B->A, Z->Y->Z など

NG例: A->C->E, A->Z->Z, Y->Z->A など

3層目に到達するとゴールになります。

各ノードには、1..26の数値(コスト)がランダムに設定されています。

1つの層の中で、コストは重複しないものとします。

例: 最小経路を通った場合、コストは $1+1+1=3$

例: 最大経路を通った場合、コストは $26+26+26=78$

ゴールするとコストが通知されます。ゴールする度に、各層毎にランダムに2ノードが選択され、コストが交換されます。

以上の動作をするサーバに対し、socket通信で「経路指示 => コスト取得」を繰り返しながら、総コストを極力最小にするようなクライアントプログラムを作成してください。

詳細なサンプルは [q5/README.txt](#) を参照して下さい。

問題 6

大貧民対戦ロボプログラムを作成してください。
ロボ同士を戦わせて順位を競います。
数千回対戦させる事で、あなたのロボの能力を数値化する事ができます。

ロボは大貧民プロトコル(以下DHMP)で通信します。

DHMP	プロトコル内容
GAME [ゲーム番号]	ゲーム開始時に通知されます。
DEAL [カード...]	カードが配られた際に通知されます。
EXCH [枚数]	前回順位が3位以外の場合、カード交換が発生します。[枚数]のカードを指定する必要があります。なお、[枚数]が負の場合、強いカードを渡さなければいけません。
REDY [プレイヤー名=カード枚数...]	準備が整ったことを通知します。
TURN [プレイヤー名] [カード情報]	[プレイヤー名]が他人の場合、そのプレイヤー出したカードが通知されます。 [プレイヤー名]が自分の場合は、出したいカードを指定して下さい。カード名を指定しなければ、パスとなります。
BIND [しぼり]	しぼりが発生した場合に通知されます。
REVO [1 or 0]	革命が発生した場合にその状態が通知されます。
FAIL [プレイヤー名] [エラー内容]	ルールにそぐわないカードを出した際に通知されます。
TEND	ターンの終了を通知します。
GEND [プレイヤー名] [順位]	順位が確定した際に通知されます。

perl版 / java版のサンプルを用意しています。

telnet コマンドからも参加できるので、まずは試してみてください。

手順については q6/perl、q6/javaそれぞれの startup.txt を参照して下さい。

【使用カード】

52枚+Joker1枚、計53枚。

マーク表記: スペード=s クラブ=c ダイヤ=d ハート=h

数字表記: A 2 3 4 5 6 7 8 9 10 J Q K

- 例1: スペードのエース => "sA"
- 例2: ダイヤの10 => "d10"
- 例3: Joker => "JKR"

【ルール】

<http://playingcards.jp/game/daifugo/rule.html>

イレブンバックのみ不採用。↑に表記されているその他ローカルルールは全て採用。

【ルール補足】

- プレイヤー人数は5人固定
- シークエンスは 2 と 3 はつながらない(例: A+2+3はNG)
- 革命はカードを出した時点で成立とする。(例:4を4枚出したら次に出せるのは3の4枚のみ)
- 革命はJokerを含めた4枚以上でも成立とする
- しばりは2人目で成立する(例: スペード4 => スペード6 となった時点でスペードしばり)
- しばりはシークエンスでも成立する
- Jokerを含んだペアやシークエンスは、しばりは成立しないが、しばられた状態を出すことはできる
- スペードの3はJokerに勝てるが、しばりによる制限は受ける
- (大)貧民が(大)富豪に渡すべきカードを渡さなかった場合、強制的に強いカードが渡される
- 10回に一度、席替えが発生する
- ルールにそぐわないカードを出した場合、そのカードは無効となり、都落ちと同じ扱いとなる