プログラミング演習 I（平成 27 年度）
– 第 1 回　レポート課題 –10/28 改訂版

担当: 時井，松村

提出日 11 月 11 日 (水) 15 時

設問 1

つくばエクスプレスにおいて，降車駅を入力すると，つくば駅から降車駅までの駅名と時間を計算し表示し，さらに乗換可能な駅名の横には [乗換可] と表示するプログラムを作成する．なお下記の条件を満たすこと．

[プログラムの条件]

- 普通列車と快速列車の停車駅を保存する配列は，下記に示す配列名と要素を使う．

```
# 普通列車
eki_futsuu = ["秋葉原", "新御徒町", "浅草", "南千住", "北千住",
"青井", "六町", "八潮", "三郷中央", "南流山",
"流山セントラルパーク", "流山おおたかの森", "柏の葉キャンパス",
"柏たなか", "守谷", "みらい平", "みどりの", "万博記念公園",
"研究学園", "つくば"]


# 快速列車
eki_kaisoku = ["秋葉原", "新御徒町", "浅草", "南千住", "北千住",
   "南流山", "流山おおたかの森", "守谷", "つくば"]
```

- 普通列車と快速列車の所要時間を保存する配列は，下記に示す配列名と要素を使う．

```
# (配列 jikan_futsuu のみかた：秋葉原と新御徒町の所要時間 (2 分) が 0 番目の要素)
# 普通列車
jikan_futsuu = [2, 2, 3, 3, 3, 2, 4, 3, 3, 3, 2, 3, 3, 4, 5, 3, 3, 3, 3]

# (配列 jikan_kaisoku のみかた：守谷とつくば駅の所要時間 (13 分) が 7 番目の要素)
# 快速列車
jikan_kaisoku = [2, 2, 3, 3, 10, 5, 7, 13]
```

- 乗換可能な駅名を保存する配列は，下記に示す配列名と要素を使う．

```
# 乗換可能駅名配列
norikae = ["秋葉原", "新御徒町", "南千住", "北千住",
"南流山", "流山おおたかの森", "守谷", "つくば"]
```

- 上記に示した 5 つの配列は，プログラム中で要素の保存順を逆順に入れ替えることは行わないこと．
- 降車駅名をキーボードから入力し，それぞれ変数に保存するキーボードから駅名を入力できるようにするためにプログラムの 1 行目の `#encoding` 文は下記のように記述してください．

```
#encoding: Windows-31J
```

- 駅名を表示するメソッド `tx_hyouji` を作成する．引数は，駅名配列，降車駅，乗換可能駅名配列とする．普通列車，快速列車共通で使用するメソッドとすること．
- 入力された駅名が駅名配列にない場合は,「降車駅名が間違っています」と表示する．もし入力された駅名が駅名配列に存在する場合は，メソッドを呼び，駅一覧と時間を表示する．
- 表示は下記に示すように配列のインデックス 駅名 `==>` の順でつくば駅から表示し，最後の降車駅では， 配列のインデックス 駅名 と表示する．乗り換えが可能な駅に関しては，駅名の後ろに [乗換可能] と表示する．
- 実行結果として，以下の 4 例を降車駅として実行した結果を示す．秋葉原，守谷，八潮，上野

```
--------- 表示結果
Z:\progI> ruby s9911999-1.rb
降車駅を入力してください: 秋葉原
19 つくば ==>
18 研究学園 ==>
17 万博記念公園 ==>
16 みどりの ==>
15 みらい平 ==>
14 守谷 [乗換可能] ==>
13 柏たなか ==>
12 柏の葉キャンパス ==>
11 流山おおたかの森 [乗換可能] ==>
10 流山セントラルパーク ==>
9 南流山 [乗換可能] ==>
8 三郷中央 ==>
7 八潮 ==>
6 六町 ==>
5 青井 ==>
4 北千住 [乗換可能] ==>
3 南千住 [乗換可能] ==>
2 浅草 ==>
1 新御徒町 [乗換可能] ==>
0 秋葉原 [乗換可能]
所要時間は，57 分です

== 快速がとまります．==
8 つくば ==>
7 守谷 [乗換可能] ==>
6 流山おおたかの森 [乗換可能] ==>
5 南流山 [乗換可能] ==>
```

```
4 北千住 [乗換可能] ==>
3 南千住 [乗換可能] ==>
2 浅草 ==>
1 新御徒町 [乗換可能] ==>
0 秋葉原 [乗換可能]
所要時間は，45 分です

Z:\progI> ruby s9911999-1.rb
降車駅を入力してください：守谷
19 つくば ==>
18 研究学園 ==>
17 万博記念公園 ==>
16 みどりの ==>
15 みらい平 ==>
14 守谷 [乗換可能]
所要時間は，17 分です

== 快速がとまります．==
8 つくば ==>
7 守谷 [乗換可能]
所要時間は，13 分です

Z:\progI> ruby s9911999-1.rb
降車駅を入力してください：八潮
19 つくば ==>
18 研究学園 ==>
17 万博記念公園 ==>
16 みどりの ==>
15 みらい平 ==>
14 守谷 [乗換可能] ==>
13 柏たなか ==>
12 柏の葉キャンパス ==>
11 流山おおたかの森 [乗換可能] ==>
10 流山セントラルパーク ==>
9 南流山 [乗換可能] ==>
8 三郷中央 ==>
7 八潮
所要時間は，38 分です

== 快速はとまりません ==

Z:\progI> ruby s9911999-1.rb
降車駅を入力してください：上野
降車駅名が間違っています．
```

プログラム作成手順の例

次のように順々にプログラムを組み立てていくとよい.

step1（駅名配列を受け取り表示するメソッドを作成）
配列の要素を後ろから順に全て改行しながら表示する.
==> と要素の間に記号を表示する.

step2
駅名の前に数字 (配列のインデックス）を表示する.
駅間の所要時間を足し合わせて，最後に表示する.

step3
降車駅を入力し，変数に保存する.
駅名が存在するかどうか判定する.

step4（駅名配列，降車駅名と乗換可能駅配列を渡すメソッドに変更する）
つくばから降車駅までの駅名を表示する

step5 表示結果例のように [乗換可能] な駅は，駅名の後ろに表示する.

さらに上を目指す方

乗車駅も入力できるようにする．その際は，メソッドの引数に乗車駅名を追加してよい.

設問２

キャラ名と能力のリストをファイルから読み込み，

- キャラのリスト，能力の平均値，最大値，
- ２つのキャラの対戦結果，

を出力するプログラムを作成せよ．

[プログラムの条件]

1. キャラの情報を格納し，特徴を表示する部分
    - キャラの能力は３つの要素「図書館愛」「コミュ力」「勤勉さ」からなる．それぞれ，０～１００の整数値になっている．また，そのキャラの総合力は，図書館愛×（コミュ力＋勤勉さ）／２の計算式で求められる整数である．
    - キャラの能力はキャラ名をキーとして要素毎に３つのハッシュに格納する．ハッシュ名は図書館愛は `toshoai`，コミュ力は `komyu`，勤勉さは `kinben` とする．
    - キャラの総合力もハッシュに格納する．ハッシュ名は `sougou` とする．
    - キャラ名と能力は，キャラ名，図書館愛，コミュ力，勤勉さの順で繰り返しキーボードから入力できるようにする．ただし，キャラ名の入力時に．(ピリオド) を入力したら終了するようにする．
    - ハッシュの値の平均値を求めるメソッド (`heikin`) を定義して使う．これは上記のハッシュを１つ受け取って，平均値を返すメソッドとする．平均値は小数点以下を切り捨てて整数とする．
    - ハッシュの値が最大のキャラを求めるメソッド (`saidai_chara`) を定義して使う．これは，上記のハッシュを１つ受け取って，値が最大のキャラ名を返すメソッドとする．
    - 全てのキャラの入力が終わったら，全キャラの能力を表示する．
    - 総合力と３つの能力それぞれに対して，平均値と最大のキャラ名とその能力の値を表示する．
2. 対戦する部分
    - ２つのキャラ名と対戦モードに対応するハッシュを受け取って，対戦結果を表示するメソッド (`taisen`) を定義して使う．対戦モードとは，図書館愛，コミュ力，勤勉さ，総合力のいずれかとする．
    - ２つのキャラのうち，ハッシュの値が大きい方が勝ちとなる．ハッシュの値が同じ場合は，引き分けとなる．
    - ２つのキャラ名と対戦モードはキーボードから繰り返し入力できるようにする．また，１つ目のキャラ名の入力時に．(ピリオド) を入力したら終了するようにする．

[実行結果の条件]

入力ファイル (chara.txt，演習ページよりダウンロードする) をリダイレクトで作成したプログラムに渡した結果を実行結果とする．

ただし，入力を促すメッセージを出力するプログラム部分（例：print("キャラ名を入れてください＞")，など）がある場合には，全てコメントアウトしてから実行すること．

[実行結果例 (データは課題とは別のもの)]

以下の例で，“:” は出力が続いていることをあらわしている記号である．

```
Z:\progI> ruby s9911999-2.rb < chara.txt
★★ キャラ一覧
キャラ名：総合力（図書館愛，コミュ力，勤勉さ）
さる：250（10，30，20）
かに：600（20，10，50）
：
：
ポニョ：3200（80，70，10）

★★ 特徴
総合力の平均は 1000 で，最大は，ＡＡの 6400 です．
図書館愛の平均は 60 で，最大は，ＸＸの 88 です．
コミュ力の平均は 30 で，最大は，ＹＹの 95 です．
勤勉さの平均は 70 で，最大は，ＺＺの 99 です．

★★ 戦いです
さる 対　かに！対戦モード：総合力
かにの勝ちです！

ポニョ　対　五郎丸！対戦モード：図書館愛
ポニョの勝ちです！

きのこの山　対　たけのこの里！対戦モード：コミュ力
引き分けです！
：
：
プログラムを終了します

Z:\progI>
```

[入力ファイル (chara.txt) の形式]（課題とは別の例です）

このファイルは，キャラと能力３つのリストに，対戦する２つのキャラ名と対戦モードのリストになっている．[プログラムの条件] にしたがって，キャラの情報の終了に . (ピリオド) が１つ，対戦情報の終了に . (ピリオド) が１つとなっている．

次ページの例は，演習ページに掲載のものとは別の例である．

```
ポニョ
80
70
10
うに
10
10
10
：
    （中略）
：
さる
10
30
20
.
さる
かに
総合力
ポニョ
五郎丸
図書館愛
：
    （中略）
：
ITF
うに
コミュ力
.
```

典型的なプログラムの構成例

```
# ハッシュの値の平均値を求めるメソッド heikin の定義

# ハッシュの値が最大のキャラを求めるメソッド saidai_chara の定義

# 対戦結果を表示するメソッド taisen の定義

# 使用するハッシュ toshoai, komyu, kinben, sougou の定義

# キャラ名と能力の値をキーボードから入力する部分

# 全キャラのキャラ名と能力の表示部分

# 総合力，3 つの要素それぞれの平均と最大を表示する部分． heikin と saidai_chara を使う

# 2 つのキャラ名と対戦モードをキーボード入力し，対戦結果を表示する部分．taisen を使う
```

提出における注意

- 締切：締切を厳守すること．締切に遅れたレポートは受理しない．
- 提出物：紙，プログラム
- 提出場所：（紙）学務課レポート提出用ポスト,（プログラム）progI-ml@klis.tsukuba.ac.jp
  自分の受講クラスのポストに入れること．他のクラスのポストに入れた場合は受理しない．
- プログラムについて
  - 作成したプログラム ***.rb を電子メールに添付して送付する．
  - ファイル名は，ユーザ名-課題番号.rb とする．すなわち，sXXXXXXX-1.rb，sXXXXXXX-2.rb の二つである．
  - 宛先は，progI-ml@klis.tsukuba.ac.jp
  - 件名は，rep1
- 紙について以下を遵守しないレポートは減点の対象となる．
  - 設問毎に A4 用紙にまとめ左上をステープラ（ホチキス）でとめる．(今回は設問が 2 個あるので，合計 2 部提出)
  - 設問毎に表紙，本文の順でとめる．
    * 表紙： 科目名，曜日クラス，第 1 回レポート　設問番号，提出日，学籍番号，氏名を記入する．見本を Web ページに掲載するので，その形式にしたがうこと．
    * 本文： 行番号付きプログラムリスト，実行結果とプログラムの説明を この順で 載せる．
      プログラムリストと実行結果は続けて記述しても良いが，プログラムの説明は別ページとすること．
      いずれも機械出力とする．その際，プログラムリストと実行結果は，等幅フォント とする．
      出力したプログラムや実行結果に手書きしないこと．
      なお，付録に形式の一例を載せたので参考にしてほしい．
  - 実行結果に関する注意事項
    * 実行は，全て全学計算機システムの Windows 上で行うこと．
    * 実行 (例 ruby ***.rb)，出力される実行結果の順となるように，一連の操作の出力を記述する．(注意：余分な操作を途中で入れないこと）
    * 実行結果は必要に応じて複数示すこと．
  - 可読性が良くなるよう努めること.「プログラム」は適切な字下げと空行の挿入をし,「説明」は適切な見出しを付けるなど，構造がわかるように注意すること．
  - 片面印刷にすること．

その他の注意

- 同一/類似レポートは「両成敗」．すなわち，見た方も見せた方も不正行為とみなし，単位を出さない．また，他の科目にも影響することもある．他人が不正にプログラムにアクセスできないよう，各自の責任において対処すること．
- レポートの差し替えや再提出は認めないので，内容をよく確認してから提出すること．
- レポートを書く上での一般常識として，参考文献がある場合には書誌情報を載せること．

- フォーマットが細かく設定されているのは決して意地悪からではない．100 名を越える受講者のレポートを効率的かつ公平に採点するために必要なことだからである．

  言い替えれば，フォーマットを無視したレポートは演習担当者の作業を著しく妨げることになる点を理解してほしい．

(付録) 行番号付きプログラム等の形式の例

以下は一例であるので，各自で考えてより読みやすい形式に整えること．

設問 1：球の体積計算プログラムを作成する．

```
----------------------------------------
プログラムリスト
----------------------------------------
 1  print("球の半径を入力してください\n")
 2  r = gets.chomp.to_f
 3  v =
 4
 .......................

----------------------------------------
実行結果
----------------------------------------
Z:\> ruby s9911999-1.rb
 .......................
 .......................

Z:\> ruby s9911999-1.rb
 .......................
 .......................
```

```
----------------------------------------
プログラムの説明
----------------------------------------
 .......................
.......................
.......................
```