

プログラミング演習I (平成23年度)

－ 第1回 レポート課題 －

担当: 時井, 松村

提出日 10月26日 (水) 15時

1 課題1

東京都内をぐるっと一周する山手線の乗車駅と降車駅を入力すると、下記の情報を表示するプログラムを作成する。

[表示する情報]

1. 乗車駅から降車駅までの駅名一覧 (外回り)
2. 所要時間 (外回り)

[プログラムの条件]

- 駅名を保存する配列に関して
 - － 山手線外回りの駅名を保存する配列を準備する。
 - － 外回りの停車駅を順に保存する配列名は、`eki_soto`とする。
 - － 東京, 有楽町, ..., 秋葉原, 神田の順に保存する。
- 所要時間を保存する配列に関して
 - － 山手線外回りの各駅間の所要時間 (分) を保存する配列を準備する。
 - － 外回りの停車駅を順に保存する配列名は、`jikan_soto`とする。
 - － 東京と有楽町、有楽町と新橋、.... 秋葉原と神田、神田と東京の所要時間の順に保存する。
- 乗車駅と降車駅を入力後、入力した駅名が配列に存在しない場合は、その旨を示すメッセージを表示する。
- 乗車駅と降車駅が配列に存在する場合は、山手線外回りの乗車駅と降車駅間の駅名リストと所要時間を表示する。
- 駅名と所要時間は下記のデータを使うこと (授業ページに掲載予定)
 - － 外回り: 駅名
["東京", "有楽町", "新橋", "浜松町", "田町",
"品川", "大崎", "五反田", "目黒", "恵比寿",
"渋谷", "原宿", "代々木", "新宿", "新大久保",
"高田馬場", "目白", "池袋", "大塚", "巣鴨",
"駒込", "田端", "西日暮里", "日暮里", "鶯谷",
"上野", "御徒町", "秋葉原", "神田"]

- 外回り所要時間(分) 東京 ↔ 有楽町, 有楽町 ↔ 新橋, 秋葉原 ↔ 神田, 神田 ↔ 東京の順

[2, 2, 2, 2, 3,
2, 2, 2, 3, 2,
3, 2, 2, 2, 3,
1, 3, 3, 1, 2,
3, 1, 2, 1, 2,
2, 2, 2, 2]

[実行結果表示例]

```
Z:\2011_progI\rep>ruby -Ks s1199999-1.rb
```

乗車駅を入力してください:東京

降車駅を入力してください:上野

[外回り]

東京 ==> 有楽町 ==> 新橋 ==> 浜松町 ==> 田町 ==> 品川 ==> 大崎 ==> 五反田 ==> 目黒 ==> 恵比
寿 ==> 渋谷 ==> 原宿 ==> 代々木 ==> 新宿 ==> 新大久保 ==> 高田馬場 => 目白 ==> 池袋 ==> 大
塚 ==> 巣鴨 ==> 駒込 ==> 田端 ==> 西日暮里 ==> 日暮里 ==> 鶯谷 ==> 上野

所要時間は 53 分です

```
Z:\progI\rep>ruby -Ks s1199999-1.rb
```

乗車駅を入力してください:東京

降車駅を入力してください:秋葉原

東京 ==> 有楽町 ==> 新橋 ==> 浜松町 ==> 田町 ==> 品川 ==> 大崎 ==> 五反田 ==> 目黒 ==> 恵比
寿 ==> 渋谷 ==> 原宿 ==> 代々木 ==> 新宿 ==> 新大久保 ==> 高田馬場 ==> 目白 ==> 池袋 ==> 大
塚 ==> 巣鴨 ==> 駒込 ==> 田端 ==> 西日暮里 ==> 日暮里 ==> 鶯谷 ==> 上野 ==> 御徒町 ==> 秋葉
原

所要時間は 57 分です

```
Z:\progI\rep>ruby -Ks s1199999-1.rb
```

乗車駅を入力してください:とうきょう

降車駅を入力してください:上野

乗車駅に指定した とうきょう は山手線の駅ではありません

```
Z:\progI\rep>ruby -Ks s1199999-1.rb
```

乗車駅を入力してください:東京

降車駅を入力してください:うえの

降車駅に指定した うえの は山手線の駅ではありません

```
Z:\progI\rep>
```

[下記のように段階をおって、課題の要件を全て満たすようにするとよい]

1. 東京から神田までの駅一覧と東京から神田までの所用時間(分)を計算し表示してみよう
2. 上野から新橋までの駅一覧と上野から新橋までの所用時間(分)を計算し表示してみよう

上野 ==> 御徒町 ==> 秋葉原 ==> 神田 ==> 東京 ==> 有楽町 ==> 新橋

3. 乗車駅と降車駅を入力し、駅名配列の何番目が求めてみよう
4. 乗車駅と降車駅を入力し、駅名一覧と所要時間を表示してみよう

2 課題2

回転寿司のお会計をするプログラムを作成する。

プログラムの条件

1. 寿司ネタデータをハッシュとして蓄積する部分

- 寿司ネタをキーとし値段をハッシュとするハッシュ(ハッシュ名 sushi)を作成する。
- 値段は整数として扱う。
- 寿司ネタ, 値段をキーボードから入力できるようにし, ハッシュsushi に蓄積する。
- 入力は while を使った繰り返し文で記述し, 寿司ネタはいくつでも入力可能とする。
- 寿司ネタの入力の終了条件は.(ピリオド) とする。
- 既にハッシュに存在する寿司ネタを指定したときは, 上書きするかそのままにするかを選択できるようにする。
- 入力が終わったら, 全ての寿司ネタと値段を表示する。
- それぞれの操作に対して, 適切なメッセージを表示する。

2. 食べた皿の数を入力し, お会計を計算する部分

- 寿司ネタを一つずつ表示し, 何皿食べたかを聞き, キーボードから皿数を入力できるようにする。
- 全部の寿司ネタについて聞き終わったら, 皿数の合計と値段の合計を出力する。

実行結果 以下の寿司ネタのデータを使って, 上記のプログラムの条件を正しく満たしていることを示す。

寿司ネタ	値段
まぐろ	190
大とろ	630
うに	525
サーモン	125
えんがわ	230
ほたて	315
はまち	230
いくら	230
あじ	315
たまご	125

3 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：印刷物，プログラムファイルの二種類
- 提出場所
 - － (印刷物) 学務課レポート提出用ポスト
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。
 - － (プログラムファイル) klis.progI@gmail.com
- プログラムファイルについて
 - － ファイル名は，ユーザ名-課題番号.rb とする。すなわち，sXXXXXXXX-1.rb，sXXXXXXXX-2.rb の二つである。
 - － 宛先は，klis.progI@gmail.com
 - － 件名は，rep1
 - － プログラムファイル2つを電子メールに添付して送付する。
- 印刷物について：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ブラ (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 x 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。

4 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的一般常識として、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名を超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 -Ks s1199999-1.rb
```

```
.....
.....
```

```
Z:\> ruby s1199999-1.rb
```

```
.....
.....
```

```
Z:\>
```

プログラムの説明

```
.....
.....
.....
```