

# プログラミング演習I (平成25年度)

## – 第2回 レポート課題 –

担当: 時井, 松村

提出〆切 平成25年11月7日(木) 15時

### 1 課題1

キーボードから野菜名と個数を受け取り、グラフにするプログラムを作成する。下記の条件を満たすこと。

[プログラムの条件]

- キーボードから受け取った野菜名と個数は、ハッシュzaiko に野菜名をキー、個数を値として保存する。
- 読み込みは while 文を使い、野菜名、個数の順に読み込み、保存する。野菜名として "." (ピリオド) が入力された場合は、読み込みを終了する。入力を促すメッセージ (「野菜名を入力して下さい」など) は今回の課題では省略する。最大 100 組の野菜名と個数を読み込むことができるよう繰り返しを設定する。
- ハッシュzaiko を引数として、メソッド hyouji を呼び、グラフを表示する。メソッド hyouji は、ハッシュzaiko のキーと値を使いグラフを表示する。グラフの形式は、野菜の個数 100 個毎に "+" (プラス記号) を表示した後に、半角スペースを 2 つ、その後 ":" (コロン) と野菜名とする。なお、100 個以下は端数として切り捨ててグラフを表示する。

以下に例を示す。

```
+++++++ :daikon
+++++++ :ninjin
+++++ :hakusai
++++ :jyagaimo
+++++++ :tomato
+++++++ :kyuuri
+++++++ :kabu
```

- 実行結果は野菜名と個数のデータが保存されたファイル (rep1\_input.txt) をリダイレクトで入力に使用して示したものとす。rep1\_input.txt は 野菜名、個数を順に保存したデータであり、授業ページに掲載するので各自がダウンロードして使うこと。
- すなわち、以下の通り実行した結果を実行結果として提出する。

```
Z:\progI> ruby rep1.rb < rep1_input.txt
```

## 2 課題2

新幹線のぞみにおいて、乗車駅名と降車駅名を入力すると乗車駅から降車駅までに通過する駅の一覧を表示するプログラムを作成する。下記の条件を満たすこと。

[プログラムの条件]

- 下記に示す駅名を保存した配列 eki を使う。

```
eki = ["東京", "品川", "新横浜", "名古屋", "京都", "新大阪", "新神戸", "姫路", "岡山",  
      "福山", "広島", "徳山", "新山口", "小倉", "博多"]
```

- 配列 eki を引数として受け取り、駅一覧を表示するメソッド ichiran を作成し、のぞみの停車駅一覧を表示する。

表示は実行例に示すように、駅名と駅名の間 ":" を表示する。ただし、博多の後ろには、 ":" を表示しないこと。

- 乗車駅名と降車駅名をキーボードから入力し、変数に保存する。
- 乗車駅名と降車駅名のどちらか一方、または両方が eki 配列に存在しない場合は、「停車駅一覧を表示できません」と表示する。

- 乗車駅、降車駅ともに、eki 配列に駅名が存在する場合は、メソッド hyouji に乗車駅名、降車駅名と配列 eki を引数として渡し、停車駅一覧を表示する。停車駅一覧を表示するメソッド hyouji は、引数として乗車駅、降車駅と配列 eki を受け取る。

表示は実行例に示すように、駅名と駅名の間 "==" を表示する。ただし、停車駅名の後ろには、 "==" は表示しない。

- 実行結果として、下記に示す4例の乗車駅と降車駅を示すこと。(東京, 博多), (名古屋, 岡山), (東京, 福岡), (上野, 仙台)

```
Z:\prog1>ruby rep2.rb
```

```
東京:品川:新横浜:名古屋:京都:新大阪:新神戸:姫路:岡山:福山:広島:徳山:新山口:小倉:博多
```

```
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:博多
```

```
東京==>品川==>新横浜==>名古屋==>京都==>新大阪==>新神戸==>姫路==>岡山==>福山==>広島==>徳山==>新山口==>小倉==>博多
```

```
Z:\prog1>ruby rep2.rb
```

```
東京:品川:新横浜:名古屋:京都:新大阪:新神戸:姫路:岡山:福山:広島:徳山:新山口:小倉:博多
```

```
乗車駅を入力してください:名古屋
```

```
降車駅を入力してください:岡山
```

```
名古屋==>京都==>新大阪==>新神戸==>姫路==>岡山
```

```
Z:\prog1>ruby rep2.rb
```

```
東京:品川:新横浜:名古屋:京都:新大阪:新神戸:姫路:岡山:福山:広島:徳山:新山口:小倉:博多
```

```
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:福岡
```

```
停車駅一覧を表示できません
```

```
Z:\progI>ruby rep2.rb
```

```
東京:品川:新横浜:名古屋:京都:新大阪:新神戸:姫路:岡山:福山:広島:徳山:新山口:小倉:博多  
乗車駅を入力してください:上野  
降車駅を入力してください:仙台  
停車駅一覧を表示できません
```

## 2.1 プログラム作成にあたって

条件を満たすプログラムを作成するにあたって、下記の手順でプログラムを作成してみるとよい。

- 配列 `eki` を作成し、要素を全て `while` 文を使って表示する。
- 駅名と駅名の間には、 ":" を表示したのち、最後の要素の後ろには、 ":" を表示しないように工夫する。
- メソッド `ichiran` を作成し、配列 `eki` を引数として受け取り駅名を表示できるようにする。
- 駅名をキーボードから 2 個入力できるようにする。
- 駅名が存在するかどうか判定できるようにする。
- 駅名が存在する場合は、停車駅一覧を表示する。
- 停車駅一覧を表示する。2 つの駅名と配列 `eki` を引数とするメソッド `hyouji` を作成する。

### 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：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）klis.progI@gmail.com  
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，sXXXXXXXX-1.rb, sXXXXXXXX-2.rb の二つである。
  - － 宛先は，klis.progI@gmail.com
  - － 件名は，rep2
- 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 2 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 rep1.rb  
.....  
.....  
  
Z:\> ruby rep1.rb  
.....  
.....
```

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