Computing for Mathematical Physics 2022/23

Homework2

Mark for homework2: 25/28
(to be completed by your marker)

Feedback from marker:
(to be completed by your marker)

Q1 8/10: Good compact answers. A few minor errors.
Q2 6/6: Very good answers. Well done.
Q3 11/12: Good. You missed a part of one question.

Well done on some excellent answers. In future worksheets please add some commentary to explain your answers. I allowed your answers to stand on their own here, and did not penalise this, but in general you should include explanations for why the code does what it does, e.g. using comments with (*) or extra text cells.*

Give your answers in the code cells marked (* Enter your solution here *)

Data for use in questions 1 and 2.

- The exercises make use of the following lists.
- *Execute the following three input cells with* SHIFT+ENTER *before attempting the questions.*
- Do not edit the data below in any way.

The following cell contains data on the planets. It will be used in question 1.

In[133]:=

```
(* Planet data entries: name, type, mean radius (km), mass (kg) *)
planetaryData = {
   {"Mercury", "Terrestrial", 2439.7, 3.3011 * 10^23},
   {"Venus", "Terrestrial", 6051.8, 4.8675 * 10^24},
   {"Earth", "Terrestrial", 6371.0, 5.9724 * 10^24},
   {"Mars", "Terrestrial", 3389.5, 6.4171 * 10^23},
   {"Jupiter", "Gas Giant", 69 911, 1.8982 * 10^27},
   {"Saturn", "Gas Giant", 58 232, 5.6834 * 10^26},
   {"Uranus", "Ice Giant", 25 362, 8.6810 * 10^25},
   {"Neptune", "Ice Giant", 24 622, 1.0241 * 10^26}
  };
```

The following code snippet creates a list containing the first 100,000 digits of π, storing it as `piDigits`. This is to be modified in question 2.

In[134]:=

```
piDigits = RealDigits[N[π, 100 000]][[1]];
```

The following function is also for use in question 2. It takes a list as input and returns `True` if the length of the list is 1, and `False` otherwise.

In[135]:=

```
lengthIsOneQ[sublist_] := (Length[sublist] == 1);
```

Questions

Double click the vertical braces on the RHS of each of the three question headings to open and view them.

- 1. Manipulating data. [10 marks]
  - The entries in `planetaryData`, in the section of the notebook above titled *Data for use in questions 1 and 2*, each comprise of a planet name, along with its type, mean radius (**km**), and mass (**kg**). In this question we will perform various manipulations on this data. Your answers should not use any manual/by-hand manipulations of the list, but instead must be carried out with *Mathematica*'s list manipulation functions.
  - a) Using `Transpose` and an appropriate value within the `[[...]]` operator, generate a new list from `planetaryData` containing only the names of the planets. Use `Sort[...]` to put the list in alphabetical order.
    [2 marks]

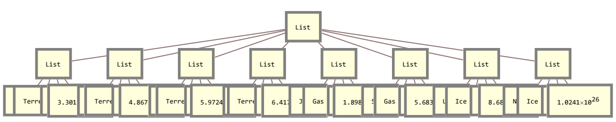
In[136]:=
```
(* Enter code in a new cell immediately below this one. *)
planetnames = Transpose[planetaryData][[1]]
Sort[planetnames]
```
Out[136]=
```
{Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune}
```
Out[137]=
```
{Earth, Jupiter, Mars, Mercury, Neptune, Saturn, Uranus, Venus}
```

Q1a 1/1

- *Good answer.*

  - b) Use `Count`, with an appropriate level-specification, to find the number of planets of each type. Present the results in a list of the form `{{"Terrestrial", no. of terrestrials}, {"Gas Giant", no. of Gas Giants}, {"Ice Giant", no. of ice giants}}`.
    [Hint: consider the output from applying `Treeform` to `planetaryData`, to help understand the correct level parameter to use with `Count`.]
    [2 marks]

In[138]:=
```
(* Enter code in a new cell immediately below this one. *)
TreeForm[planetaryData]
gg = Count[planetaryData, "Gas Giant", {2}]
ig = Count[planetaryData, "Ice Giant", {2}]
tr = Count[planetaryData, "Terrestrial", {2}]
planettypes = {{"Terrestrial", tr}, {"Terrestrial", tr}, {"Ice Giant", ig}}
```
Out[138]//TreeForm=



Out[139]=
```
2
```
Out[140]=
```
2
```
Out[141]=
```
4
```
Out[142]=
```
{{Terrestrial, 4}, {Terrestrial, 4}, {Ice Giant, 2}}
```
In[143]:=
```
{{"Terrestrial", 4}, {"Ice Giant", 2}, {"Terrestrial", 4}} // MatrixForm
```
Out[143]//MatrixForm=

$$\begin{pmatrix} \text{Terrestrial} & 4 \\ \text{Ice Giant} & 2 \\ \text{Terrestrial} & 4 \end{pmatrix}$$

Q1b 1/2

- *You had the right basic idea, but you made a typo in your definition of planettypes, so the Gas Giant count was omitted.*

- c) Sort the entries in `planetaryData` from smallest to largest *mass*. Do this by first using `RotateRight`, with an appropriate level specification, to shuffle the data structure such that the masses appear as the first column, then apply `Sort`, and finally *undo* the initial shuffling with `RotateRight` by an appropriate call to `RotateLeft`. [2 marks]

In[144]:=

```
(* Enter code in a new cell immediately below this one. *)
planetmass1 = RotateRight[planetaryData, {0, 2}]
planetmass2 = Sort[planetmass1]
planetmass3 = RotateLeft[planetmass2, {0, 2}]
```

Out[144]=

{{2439.7, 3.3011×10^{23}, Mercury, Terrestrial},
{6051.8, 4.8675×10^{24}, Venus, Terrestrial},
{6371., 5.9724×10^{24}, Earth, Terrestrial},
{3389.5, 6.4171×10^{23}, Mars, Terrestrial},
{69 911, 1.8982×10^{27}, Jupiter, Gas Giant},
{58 232, 5.6834×10^{26}, Saturn, Gas Giant},
{25 362, 8.681×10^{25}, Uranus, Ice Giant},
{24 622, 1.0241×10^{26}, Neptune, Ice Giant}}

Out[145]=

{{2439.7, 3.3011×10^{23}, Mercury, Terrestrial},
{3389.5, 6.4171×10^{23}, Mars, Terrestrial},
{6051.8, 4.8675×10^{24}, Venus, Terrestrial},
{6371., 5.9724×10^{24}, Earth, Terrestrial},
{24 622, 1.0241×10^{26}, Neptune, Ice Giant},
{25 362, 8.681×10^{25}, Uranus, Ice Giant},
{58 232, 5.6834×10^{26}, Saturn, Gas Giant},
{69 911, 1.8982×10^{27}, Jupiter, Gas Giant}}

Out[146]=

{{Mercury, Terrestrial, 2439.7, 3.3011×10^{23}},
{Mars, Terrestrial, 3389.5, 6.4171×10^{23}},
{Venus, Terrestrial, 6051.8, 4.8675×10^{24}},
{Earth, Terrestrial, 6371., 5.9724×10^{24}},
{Neptune, Ice Giant, 24 622, 1.0241×10^{26}},
{Uranus, Ice Giant, 25 362, 8.681×10^{25}},
{Saturn, Gas Giant, 58 232, 5.6834×10^{26}},
{Jupiter, Gas Giant, 69 911, 1.8982×10^{27}}}

Q1c 2/2

- *Correct*

- d) Use `Transpose`, or alternatively the `Span` operator '`;;`', to extract three separate lists (columns) from the `planetaryData` data structure: `planetaryNames`,

comprising just the names of the planets, `planetaryRadii` comprising just their radii (km), and `planetaryMasses` comprising just their masses (kg). Using the latter three lists, construct a new list of the form `{{planet name 1, density 1}, {planet name 2, density 2}, ...}`, where the density should be given in $\boldsymbol{g/cm^3}$, according to the formula $M_{\text{planet}}/(\frac{4}{3}\pi R_{\text{planet}}{}^3)$, wherein M_{planet} denotes a planet's mass in g, and R_{planet} its radius in cm.
[4 marks]

In[147]:=
```
(* Enter code in a new cell immediately below this one. *)
planetaryNames = planetaryData[[1 ;; 7, 1]]
planetaryRadii = planetaryData[[1 ;; 7, 3]]
planetaryMasses = planetaryData[[1 ;; 7, 4]]
```
Out[147]=
```
{Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus}
```
Out[148]=
```
{2439.7, 6051.8, 6371., 3389.5, 69911, 58232, 25362}
```
Out[149]=

$\{3.3011\times10^{23}, 4.8675\times10^{24}, 5.9724\times10^{24}, 6.4171\times10^{23}, 1.8982\times10^{27}, 5.6834\times10^{26}, 8.681\times10^{25}\}$

In[150]:=
```
Table[{planetaryNames[[x]], 1000
   planetaryMasses[[x]] / ((4 / 3) Pi ((10^5 planetaryRadii[[x]]) ^3))}, {x, 1, 7}]
```
Out[150]=
```
{{Mercury, 5.42701}, {Venus, 5.2428}, {Earth, 5.51363}, {Mars, 3.93408},
 {Jupiter, 1.32622}, {Saturn, 0.687123}, {Uranus, 1.27037}}
```

Q1d 3 /4

- *Basically correct but Neptune should not have been excluded.*

- 2. Searching for and analysing sequences of identical elements in a list. [6 marks]

  - a) Modify the code used to create a list of the first 100,000 digits in π, found in the section of the notebook above titled *Data for use in questions 1 and 2*, to obtain a new list, `eulerDigits`, containing the first 100,000 digits in Euler's constant, 0.577216... , known to *Mathematica* as `EulerGamma`. You can ignore the leading zero in Euler's constant, i.e. your list should contain 100,000 elements, the first of which is 5. *Suppress the full list output by terminating any relevant commands with a semi-colon.* Display instead just the first 8 elements with `eulerDigits`[[1;;8]].
  [1 mark]

In[151]:=
```
(* Enter code in a new cell immediately below this one. *)
eulerDigits = RealDigits[N[EulerGamma, 100000]][[1]];
eulerDigits[[1 ;; 8]]
```
Out[152]=
```
{5, 7, 7, 2, 1, 5, 6, 6}
```

Q2a 1/1

- *Correct*

  - b) Use `Partition` with `eulerDigits` to create a new list `partitionedEulerDigits`, comprised of 99,998 three-element sublists, starting { {5,7,7}, {7,7,2}, {7,2,1}, {2,1,5}, {1,5,6}, ... }. I.e. the partitioning of `eulerDigits` is such that the second & third numbers in each sublist, reappear again as the first & second numbers in the following sublist. *Suppress the full list output by terminating any relevant commands with a semi-colon*. Display instead just the first 6 elements with `partitionedEulerDigits`[[1;;6]].
    [1 mark]

In[153]:=

```
(* Enter code in a new cell immediately below this one. *)
partitionedEulerDigits = Partition[eulerDigits, 3, 1];
partitionedEulerDigits[[1 ;; 6]]
```

Out[154]=

```
{{5, 7, 7}, {7, 7, 2}, {7, 2, 1}, {2, 1, 5}, {1, 5, 6}, {5, 6, 6}}
```

Q2b 1/1

- *Correct*

  - c) Using `Map[...]` apply `Union[...]` to each of the three-element sublists in `partitionedEulerDigits`, to create a new list called `partitionedEulerDigitsReduced`. In this new list any three-element sublists made of three identical digits in `partitionedEulerDigits` will have been replaced by one-element sublists. Similarly, any three-element sublists in `partitionedEulerDigits` containing two identical digits will have been replaced by two-element sublists. *Suppress any verbose list output by terminating any relevant commands with a semi-colon.* Instead, display just the first 6 elements of the result with `partitionedEulerDigitsReduced`[[1;;6]].
    [1 mark]

In[155]:=

```
(* Enter code in a new cell immediately below this one. *)

partitionedEulerDigitsReduced = Map[Union, partitionedEulerDigits];
partitionedEulerDigitsReduced[[1 ;; 6]]
```

Out[156]=

```
{{5, 7}, {2, 7}, {1, 2, 7}, {1, 2, 5}, {1, 5, 6}, {5, 6}}
```

Q2c 1/1

- *Good*

  - d) The function `lengthIsOneQ` defined in the section of the notebook above titled *Data for use in questions 1 and 2*, returns `True` when passed a (sub)list of length 1, and `False` otherwise. Use `lengthIsOneQ` with `Select[...]` to produce a list of all sublists of length 1 residing in `partitionedEulerDigitsReduced` (suppress the full list output with a semi-colon). Hence, determine the number of times that three consecutive digits are the same, in the first 100,000 digits of Euler's constant.
    [1 mark]

In[157]:=

```
(* Enter code in a new cell immediately below this one. *)
consecutives = Select[partitionedEulerDigitsReduced, lengthIsOneQ];
Length[consecutives]
```

Out[158]=

962

Q2d 1/1

- *Correct*

  - e) Generalise parts b), c), d) to determine the number of times that *six* consecutive digits are the same, in the first 100,000 digits of Euler's constant.
    [2 marks]

In[159]:=

```
(* Enter code in a new cell immediately below this one. *)
partitionedEulerDigits6 = Partition[eulerDigits, 6, 1];
partitionedEulerDigitsReduced6 = Map[Union, partitionedEulerDigits6];
consecutives6 = Select[partitionedEulerDigitsReduced6, lengthIsOneQ];
Length[consecutives6]
```

Out[162]=

2

Q2e 2/2

- *Nice compact code*

- 3. Basic combinatorial analysis using lists with `Outer`, `Flatten`, `Map`, `Union`, `Count`, etc.
  [12 marks]

  - a) Use `Outer`, `List`, and `Flatten`, together with the list `{1,2,3,4,5,6,7,8}` to generate a new list, `possibleOutcomes`, which shows the possible outcomes of throwing three **eight-sided** dice, in the form `{{1,1,1},{1,1,2},...}`. Note, `Outer` can take four input arguments, e.g. `Outer[Times,{a,b},{c,d},{e,f}]`.

    Suppress verbose list output by terminating any relevant commands with a semi-colon. Display instead just the first six elements with `possibleOutcomes`[[1;;6]].

    How many possibilities are there? Store this number as `numberOfPossibleOutcomes`.
    [2 marks]

In[163]:=

```
(* Enter code in a new cell immediately below this one. *)
possibleOutcomes1 = Outer[List, {1, 2, 3, 4, 5, 6, 7, 8},
   {1, 2, 3, 4, 5, 6, 7, 8}, {1, 2, 3, 4, 5, 6, 7, 8}];
possibleOutcomes = Partition[Flatten[possibleOutcomes1 ], 3];
possibleOutcomes[[1 ;; 6]]
```

Out[165]=

```
{{1, 1, 1}, {1, 1, 2}, {1, 1, 3}, {1, 1, 4}, {1, 1, 5}, {1, 1, 6}}
```

Q3a 1/2

- *Good but you failed to create the `numberOfPossibleOutcomes`. It looks like you have* *copy pasted here (hence the blue text). Don't do that. It can add extra formatting data which can corrupt your calculations.*

- b) Use `Map[...]` with `Total[...]` and `possibleOutcomes` to create a list, `possibleTotals`, where each element is given by the sum of the three dice-rolls in the corresponding element of `possibleOutcomes`. Suppress verbose list output by terminating any relevant commands with a semi-colon. Display instead just the last six elements with `possibleTotals`[[-6;;-1]].
  [1 mark]

In[166]:=

```
(* Enter code in a new cell immediately below this one. *)

possibleTotals = Map[Total, possibleOutcomes];
possibleTotals[[-6 ;; -1]]
```

Out[167]=

```
{19, 20, 21, 22, 23, 24}
```

Q3b 1/1

- *Good*

- c) Use `Union[...]` to create a list of all distinct possible totals, `distinctPossibleTotals`, from `possibleTotals`, sorted in increasing order.
  [1 mark]

In[168]:=

```
(* Enter code in a new cell immediately below this one. *)
distinctPossibleTotals = Sort[Union[possibleTotals]]
```

Out[168]=

```
{3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24}
```

Q3c 1/1

- *Good compact answer.*

- d) Use `Table[...]` with `Count[...]`, `distinctPossibleTotals`, `possibleTotals`, and `numberOfPossibleOutcomes` to determine the probability of obtaining each possible total score, as a list called, `distinctPossibleTotalsProbabilities`, of the form,

  { {score #1, prob. score #1}, {score #2, prob. score #2}, {score #3, prob. score #3}, ... }.

  [2 marks]

In[169]:=
```
(* Enter code in a new cell immediately below this one. *)
distinctPossibleTotalsProbabilities =
 Table[{distinctPossibleTotals[[x]], Count[possibleTotals, x + 2] /
    Length[possibleTotals]}, {x, 1, Length[distinctPossibleTotals]}]
```

Out[169]=

$$\left\{\left\{3, \frac{1}{512}\right\}, \left\{4, \frac{3}{512}\right\}, \left\{5, \frac{3}{256}\right\}, \left\{6, \frac{5}{256}\right\}, \left\{7, \frac{15}{512}\right\}, \left\{8, \frac{21}{512}\right\}, \left\{9, \frac{7}{128}\right\}, \left\{10, \frac{9}{128}\right\}, \left\{11, \frac{21}{256}\right\}, \left\{12, \frac{23}{256}\right\}, \left\{13, \frac{3}{32}\right\}, \left\{14, \frac{3}{32}\right\}, \left\{15, \frac{23}{256}\right\}, \left\{16, \frac{21}{256}\right\}, \left\{17, \frac{9}{128}\right\}, \left\{18, \frac{7}{128}\right\}, \left\{19, \frac{21}{512}\right\}, \left\{20, \frac{15}{512}\right\}, \left\{21, \frac{5}{256}\right\}, \left\{22, \frac{3}{256}\right\}, \left\{23, \frac{3}{512}\right\}, \left\{24, \frac{1}{512}\right\}\right\}$$

Q3d 2/2

- *Good compact method*

  - e) Confirm that the sum of all the probabilities associated to each possible total score is 1. [2 marks]

In[170]:=
```
(* Enter code in a new cell immediately below this one. *)
Total[distinctPossibleTotalsProbabilities[[1 ;; 22, 2]]]
```

Out[170]=

1

Q3e 2/2

- *Good*

  - f) In an alternative version of the game above, we throw three dice but only take the sum of the two highest numbers as our score. Calculate the probability of each of the total possible scores (from 2 to 16). This computation can be carried out more-or-less as in steps b)→e), with an adaptation to the computation of the `possibleTotals` in part b). [4 marks]

In[171]:=

```
(* Enter code in a new cell immediately below this one. *)

twohighestnumbersoutcomesinitial = Table[
   Delete[Sort[possibleOutcomes[[x]]], 1], {x, 1, Length[possibleOutcomes]}];
twohighestnumbersoutcomes =
  Partition[Flatten[twohighestnumbersoutcomesinitial], 2];
possibleTotals2highest = Map[Total, twohighestnumbersoutcomes];
distinctPossibleTotals2highest =
  Sort[Union[possibleTotals2highest]];
Table[{distinctPossibleTotals2highest[[x]],
  Count[possibleTotals2highest, x + 1] / Length[possibleTotals2highest]},
 {x, 1, Length[distinctPossibleTotals2highest]}]
```

Out[175]=

$$\left\{\left\{2, \frac{1}{512}\right\}, \left\{3, \frac{3}{512}\right\}, \left\{4, \frac{7}{512}\right\}, \left\{5, \frac{3}{128}\right\}, \left\{6, \frac{19}{512}\right\}, \left\{7, \frac{27}{512}\right\}, \left\{8, \frac{37}{512}\right\}, \left\{9, \frac{3}{32}\right\}, \left\{10, \frac{29}{256}\right\}, \left\{11, \frac{63}{512}\right\}, \left\{12, \frac{1}{8}\right\}, \left\{13, \frac{15}{128}\right\}, \left\{14, \frac{13}{128}\right\}, \left\{15, \frac{39}{512}\right\}, \left\{16, \frac{11}{256}\right\}\right\}$$

Q3f 4/4

- *Good compact code.*

- **Total marks available: 28**

- **Solutions are due *by* 1200 noon on Thursday January 26th *here*: allow time for uploading on moodle.**
- A 10% mark deduction will be made (3 marks) if the template isn't used.
- Name your solution notebook file in the format **WK2_HMWK_<Initials>_<Family Name>.nb**, e.g. **WK2_HMWK_K_Hamilton.nb**
- Make a *backup copy* of your solutions.
- Delete all cell evaluation output by selecting **Cell → Delete All Output** from the drop-down menus at the top of the screen, then save and upload *that* file to Moodle.
- The first thing your marker will do when they receive your notebook is to evaluate all of it, to regenerate the output, by clicking **Evaluation → Evaluate Notebook** from the drop-down menus at the top of the screen. *It is your responsibility to check that carrying out this process will produce the output you intend it to, before you upload your work.*

K. Hamilton
Based on J. Bhamrah, J. Underwood, L. McKemmish — UCL
Last revision 11:18 17 Jan 2023