Computing for Mathematical Physics 2022/23

Homework 3

Mark for homework 3: 25/33

(to be completed by your marker)

Feedback from marker:

A very good attempt with a clear understanding of how to think about problems in Mathematica (for example by using the Head[] function).
All of your lost marks were from the final two questions, and I've tried to explain how to solve these problems by building on your current solutions.
In all cases you have shown that you understand how to think about these problems, keep up the good work!
I have noted in a few places some ways that you can improve the readability of your code, for example by using camel case when naming your variables.
Before submission, remember to delete your outputs before submission (Cell, Delete All Output).

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

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

- 1. Substituting solutions from `Solve` using `ReplaceAll` (`/.`). [4 marks]
  - Solve the following pair of *simultaneous* equations for x and y:

    $x + \sqrt{y} = 6$ and $\text{Log}[x] + \text{Log}[y] = 4\,\text{Log}[2]$.

    Confirm that your solutions are valid by substituting them back into the original equations (above), and using, e.g, `Simplify` or `FullSimplify` *as appropriate*. [4 marks]

In[7]:=
```
eq1 = x + y^.5 == 6
eq2 = Log[x] + Log[y] == 4 Log[2]
sol1 = Solve[{eq1, eq2}, {x, y}]
```

Out[7]= $x + y^{0.5} == 6$

Out[8]= Log[x] + Log[y] == 4 Log[2]

Out[9]= {{x → 4., y → 4.}, {x → 0.535898, y → 29.8564}}

In[10]:=
```
{eq1, eq2} /. sol1[[1]]
{eq1, eq2} /. sol1[[2]]
```

Out[10]= {True, True}

Out[11]= {True, True}

4/4: Great work! It is worth noting that *Solve[]* will generate decimal outputs when decimal inputs are used (for example the *.5* power in eq1), and fractional/surd outputs otherwise. Fractional/surd outputs are preferred as they are more accurate, so try to use the fractions and the *Sqrt[]* function instead of decimals when possible in the future. Also, both substitutions of *sol1* into *eq1* & *eq2* can be completed at the same time using *{eq1, eq2}/.sol1*.

- 2. Alternative implementation of series expansion using rule-based replacement. [9 marks]
  - It may help to quickly refresh your memory on Taylor series expansion before starting here.
  - a) Use `Table[...]` and `D[...]` to construct a list containing `Log[1 − x]`, and its first three derivatives with respect to x.
    [2 marks]

In[12]:=
```
logseries3 = Table[D[Log[1 - x], {x, n}], {n, 0, 3}]
```

Out[12]= $\left\{\text{Log}[1-x], -\frac{1}{1-x}, -\frac{1}{(1-x)^2}, -\frac{2}{(1-x)^3}\right\}$

2/2: This is another good solution, but try to use the camel case notation when naming variables so that it is easier to understand your code when it is read or when you come back to it. For example, the variable *logseries3 -> logSeries3*.

  - b) Use a **rule** to convert this to a second list giving the values of `Log[1 − x]` and its derivatives at *x*=0.
    [1 marks]

In[14]:=
```
xat0 = logseries3 /. x → 0
```

Out[14]= {0, −1, −1, −2}

1/1: Nice solution!

  - c) Use `Table[...]` to construct another list containing $x^n / n!$ for n = 0, 1, 2, 3.
    [2 marks]

In[15]:=
```
taylorseries = Table[x^n / n!, {n, 0, 3}]
```

Out[15]= $\left\{1, x, \frac{x^2}{2}, \frac{x^3}{6}\right\}$

2/2: Good use of superscript and factorial notation!

- d) Combine your results from parts b) and c), to yield a series expansion of `Log[1 − x]` about `x=0`, for all terms up to and *including* terms proportional to x^3 .
  [2 marks]

In[16]:=
```
fulltaylor = xat0 taylorseries
fulltaylorsimple =
 {fulltaylor[[1]] + fulltaylor[[2]] + fulltaylor[[3]] + fulltaylor[[4]]}
```

Out[16]= $\left\{0, -x, -\frac{x^2}{2}, -\frac{x^3}{3}\right\}$

Out[17]= $\left\{-x - \frac{x^2}{2} - \frac{x^3}{3}\right\}$

2/2: Good job noticing that the product of your answers in parts (b) and (c) generate a Taylor series! This solution can be made more general by using the *Total[]* function instead of manually summing the terms of the list. The solution also does not need to be written as a list (enclosed in curly braces *({})*) since it is a single equation, but this does not affect your result.

- e) Compute the same series expansion directly using *Mathematica*'s built-in `Series` function. Appropriately test whether the latter result agrees with the one obtained in part d) using the `Equal` operator, `==`.
  [2 marks]

In[20]:=
```
trueseries = Normal[{Series[Log[1 - x], {x, 0, 3}]}]
```

Out[20]= $\left\{-x - \frac{x^2}{2} - \frac{x^3}{3}\right\}$

In[21]:=
```
trueseries == fulltaylorsimple
```

Out[21]= True

2/2: Good solution! In general the curly braces are not needed inside the *Normal[]* function as the solution does not need to be in a list, but in this case they are necessary to match the form of your solution to part (d) when comparing them using the *Equal* operator (==).

- 3. Replacement rules with refined pattern matching — conditional pattern matching `(/;)`.
  [6 marks]
  - *In answering this question, suppress any long list output (more than 10 elements) by terminating any relevant commands with a* ***semi-colon****. To display a list as part of your submitted work, limit yourself to showing the first 10 elements, by appending* `[[1;;10]]`.
  - Enter and evaluate

`goldenRatioDigits=RealDigits[N[GoldenRatio,1000]][[1]];`
to generate a list of the first 1000 digits of the so-called golden ratio.

To avoid potential problems type-in the line above, rather than cut-and-paste it.
The command is terminated by a semi-colon, indicating that *you are not supposed to display it*.

- a) Write a replacement rule to act on the latter list of digits adhering to the following terms. On the LHS of the replacement rule the pattern should match a ***list*** consisting of the following pieces

  - *any number of unspecified elements, or zero elements, followed by,*
  - *a single element* ***that is even****, followed by,*
  - *any number of unspecified elements, or zero elements.*

  You must implement the requirement that the single element, in the middle of the pattern, be an even number, by appropriately attaching a condition to the pattern, using `/;` and `EvenQ[...]`. The RHS of your replacement rule should be constructed such that the original list is returned, but with the single even element deleted.

  Apply your replacement rule repeatedly to the `goldenRatioDigits` list to delete all even digits (suppress the output with a semi-colon). Hence, display the *number* of odd digits in the first 1000 digits of the golden ratio.
  [3 marks]

In[22]:=
```
goldenRatioDigits = RealDigits[N[GoldenRatio, 1000]][[1]];
oddDigitsLength =
 Length[goldenRatioDigits //. {a___, b_, c___} /; EvenQ[b] → {a, c}]
```

Out[23]= 484

3/3: Great answer! It is good practice to store your intermediate results (for example your replacement rule and the list of odd digits generated when you use the rule on the *goldenRatioDigits* list) as separate variables in case you need them again in the future.

- b) Adapt your code for part a) to compute and display the *number* of *even* digits in the first 1000 digits of the golden ratio.
  [1 mark]

In[24]:=
```
evenDigitsLength =
 Length[goldenRatioDigits //. {a___, b_, c___} /; OddQ[b] → {a, c}]
```

Out[24]= 516

1/1: Good work adapting your method from part (a)!

- c) Use `Select` with `EvenQ`, `OddQ` and `Length` to check your results from parts a) and b).

In[25]:=
```
Length[Select[goldenRatioDigits, EvenQ]] == evenDigitsLength
Length[Select[goldenRatioDigits, OddQ]] == oddDigitsLength
```

Out[25]= True

Out[26]= True

2/2: A very neat and concise method of comparison!

- 4. Conditional replacement based on object type and value. [8 marks]
  - Write a rule, or list of rules, which replaces any objects explicitly with head type `Real`, `Integer` or `Rational`, by the same object (i.e. number), if it was positive, and the same object (i.e. number) *with it's sign flipped*, if it was negative. All other objects which are not of those types should be left unchanged. Your solution must use conditional pattern matching (`/;`) and replacement. Your solution is not permitted to use the `Abs[]` function.

    Test your rule on the following examples:
    - - 666
    - - 101.0
    - x - 100
    - - 23/45
    - $2\,x^2 - 5\,x^{-3.5}$
    - {-1.0, 50.6, -4, {7, -34.5} }

```
Head[-666]
Head[-101.0]
Head[x - 100]
Head[-23 / 45]
Head[2 x^2 - 5 x^-3.5]
```

In[38]:=
```
integer = a_ /; Head[a] == Integer → a ;
real = a_ /; Head[a] == Real → a;
rational = a_ /; Head[a] == Rational → a ;
integerRule = a_ /; a < 0 → -a ;
realRule = a_ /; a < 0 → -a;
rationalRule = a_ /; a < 0 → -a;
```

In[44]:=
```
-666 /. integerRule
-101.0 /. realRule
{x - 100 /. integerRule, x - 100 /. realRule, x - 100 /. rationalRule}
{2 x^2 - 5 x^-3.5 /. integerRule,
 2 x^2 - 5 x^-3.5 /. realRule, 2 x^2 - 5 x^-3.5 /. rationalRule}
```

Out[44]= 666

Out[45]= 101.

Out[46]= {100 + x, 100 + x, 100 + x}

Out[47]= $\{2 x^2 + 5 x^{3.5}, 2 x^2 + 5 x^{3.5}, 2 x^2 + 5 x^{3.5}\}$

3/8: A good attempt, and a great use of the *Head[]* function to try and get to a solution! This question was looking for a single list of rules that could be applied to each of the examples and generate the correct solution (with the sign flipped for any negative real numbers, integers or rational numbers). In this case, the rules you have written in the second cell of your answer need to be combined to give one list of rules. The rules you have used in the final cell of your answer are currently all the same, and do not check whether the numbers are real, rational or integers, but this can be achieved by combining all of the rules you have written:

```
combinedRule = {a_ /; (Head[a] == Integer && a < 0) → -a,
   a_ /; (Head[a] == Real && a < 0) → -a, a_ /; (Head[a] == Rational && a < 0) → -a}
```

- 5. Complex rule replacements. [6 marks]
  - **First**, generate a list of 50 random integers, stored as `q5List`, by *writing* and executing the following *two* commands in an input cell:

```
SeedRandom[1234];
q5List=RandomInteger[{0,1000},50]
```

  - a) Write a *single* compact *replacement rule* which, when applied with `ReplaceRepeated` (`//.`), sorts `q5List` into *ascending* order. Use the `Equal` operator (==) to show that your result agrees with that obtained using the native *Mathematica* sort function: `Sort[q5List]`.

    Note: your rule should implement a pattern on the LHS with a condition (`/;`). [3 marks]

In[48]:=
```
SeedRandom[1234];
q5List = RandomInteger[{0, 1000}, 50]
```

Out[49]=
```
{897, 662, 641, 768, 529, 133, 637, 815, 531, 88, 389,
 530, 756, 238, 688, 870, 47, 743, 269, 918, 444, 47, 712, 900,
 80, 708, 949, 533, 45, 573, 969, 651, 206, 587, 706, 149, 542,
 693, 491, 905, 878, 403, 947, 820, 683, 983, 40, 546, 859, 69}
```

In[50]:=
```
sort = {c___, a_, b_, d___} /; a > b → {c, b, a, d};
sortedq5List = q5List //. sort
```

Out[51]=
```
{40, 45, 47, 47, 69, 80, 88, 133, 149, 206, 238, 269, 389,
 403, 444, 491, 529, 530, 531, 533, 542, 546, 573, 587, 637,
 641, 651, 662, 683, 688, 693, 706, 708, 712, 743, 756, 768,
 815, 820, 859, 870, 878, 897, 900, 905, 918, 947, 949, 969, 983}
```

In[52]:=
```
sortedq5List == Sort[q5List]
```

Out[52]=
```
True
```

3/3: Great use of single and triple underscores in your replacement rule!

- b) Write a replacement rule which, when applied to `q5List` with `ReplaceRepeated` (`//.`), returns a list comprised of just one element, with that one element being the *largest* element in the original list `q5List`. Compare the latter result to that obtained using `Max[q5List]`.

  Your rule should comprise of *two* conditional patterns (`/;`) on the LHS combined with the `Alternatives` operator, denoted by a single vertical line, `|`. You may *not* use an explicit loop construct, or `Map`, to iterate through the list elements.

  You may wish to refer back to the end of lecture 3, and/or read into `??Alternatives`.
  [3 marks]

In[59]:=
```
max = {a___, b_, c_, d_, e___}; c < d → {a, b, d, e};
mo = {m___, r_, a___, x_, e___, s_, n___} /; x < s → {m, r, a, e, s, n}
mo2 = {m___, r_, a___, x_ /; x < r_, e___, s_, n___} → {m, r, a, e, s, n}
q5List //. mo
q5List //. mo2
q5List
```

Out[60]=
```
{m___, r_, a___, x_, e___, s_, n___} /; x < s → {m, r, a, e, s, n}
```

Out[61]=
```
{m___, r_, a___, x_ /; x < r_, e___, s_, n___} → {m, r, a, e, s, n}
```

Out[62]=
```
{897, 983, 859, 69}
```

Out[63]=
```
{897, 662, 641, 768, 529, 133, 637, 815, 531, 88, 389,
 530, 756, 238, 688, 870, 47, 743, 269, 918, 444, 47, 712, 900,
 80, 708, 949, 533, 45, 573, 969, 651, 206, 587, 706, 149, 542,
 693, 491, 905, 878, 403, 947, 820, 683, 983, 40, 546, 859, 69}
```

Out[64]=
```
{897, 662, 641, 768, 529, 133, 637, 815, 531, 88, 389,
 530, 756, 238, 688, 870, 47, 743, 269, 918, 444, 47, 712, 900,
 80, 708, 949, 533, 45, 573, 969, 651, 206, 587, 706, 149, 542,
 693, 491, 905, 878, 403, 947, 820, 683, 983, 40, 546, 859, 69}
```

0/3: Good attempt at the replacement rule, but the rules you have used include too many variables on the LHS (only two variables need to be checked, not three). The first rule you

have tried is close to working correctly, for example the rule:
`max = {a___, b_, c_, d___} /; c < b → {a, b, d}`, removes an element of the list if the right-hand element is less than the left-hand element being considered. This then needs to be paired with a rule checking in the other direction using the *Alternatives* operator:
`max = {a___, b_, c_, d___} | {a___, c_, b_, d___} /; c < b → {a, b, d};`

- **Total marks available: 33**

- **Solutions are due *by* 1200 noon on Thursday February 2nd *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 **WK3_HMWK_<Initials>_<Family Name>.nb**, e.g. **WK3_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.
Last revision 12:11 21 Jan 2023