- Assignments should be completed individually.
- No late assignments will be accepted.
- Provide **concise** answers to the following questions. Use **point form** whenever possible.
- Submit your completed solutions to **Crowdmark**.

1. **Definition 1** *a* ***Hoare triple*** *is a triple* $(\!|P|\!)\ C\ (\!|Q|\!)$ *composed of*
   - $P$*, a* ***precondition*** *(an assertion),*
   - $C$*, some code, and*
   - $Q$*, a* ***postcondition*** *(another assertion).*

   **Definition 2** *A* ***specification*** *of a program* $C$ *is a Hoare triple with* $C$ *as the middle element of the triple.*

   **Definition 3** *A Hoare triple is* ***satisfied under partial correctness*** *if, whenever execution starts in a* ***state*** *satisfying precondition* $P$*, and terminates, it follows that the* ***state*** *after execution satisfies postcondition* $Q$*.*

   **Definition 4** *The* ***state*** *of a program at a given moment is the list of the values of each of its variables at that moment.*

   For each specification below, either
   - Give an informal argument for why the specification is satisfied under partial correctness, or
   - Give an example of a starting state which demonstrates that the specification is **not** satisfied under partial correctness, and briefly explain why your choice is correct.

[4] (a) $(\!|2 + y \geq 4|\!)$

```
x = 2;
```

$(\!|x + y \geq 4|\!)$

[4] (b) $(\!|x + y \geq 4|\!)$

```
x = 2;
```

$(\!|2 + y \geq 4|\!)$

2. It is mentioned in the Lecture Notes that **cost-benefit analysis** can be used to justify proving a software product is correct when human lives depend on correctness. One ethically problematic point about this practice is that it requires assigning a dollar value to a human life. In each of the following cases, assume that human lives depend on the correctness of the software product, and that your software professionals earn $2000 / week.

[2] (a) It is estimated that a new software product will require 12 person-weeks from a software professional for proving its correctness. The IT team manager authorizes spending the time to prove this software product is correct. What is the minimum dollar value of a human life in this cost-benefit analysis? Show your work.

[2] (b) It is estimated that another new software product will require 15 person-weeks from a software professional for proving its correctness. The IT team manager does not authorize spending the time to prove this software product is correct. What is the maximum dollar value of a human life in this cost-benefit analysis? Show your work.

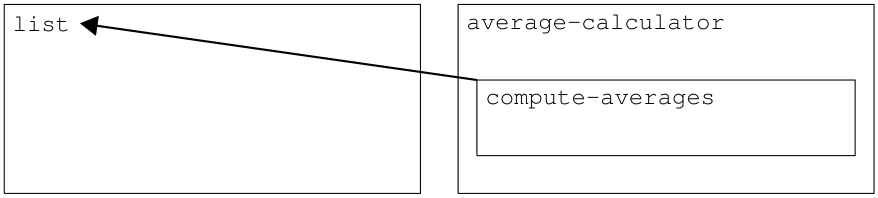
3. In each part of this question, you will evaluate a software product, to assess the **cohesion** and **coupling** of its modules. The software product computes the **mean**, **median** and **mode** of a set of examination scores. Each software product is composed of two modules. In each part of the question,
   i. state whether the modules described have **high** or **low** cohesion, and briefly justify your answer, and
   ii. state whether the pair of modules described have **loose** or **tight** coupling, and briefly justify your answer.

[4] (a) This solution uses two classes, with the given properties/methods.

i. `list`
   A. `property:  mean`
   B. `property:  median`
   C. `property:  mode`

ii. `average-calculator(my-list)`

   A. `method:  compute-averages`
```
                  /* return the triple
                  * (my-list.mean,my-list.median,my-list.mode)
                  */
```



[4] (b) This solution uses two modules, with the given functions and descriptive comments.

i. `average-calculator-1`

A. `total-entries(alist)`

```
/* sum all entries in alist by calling
* average-calculator-2.count-by-value(alist).
*/
```

B. `sort-entries(alist)`

```
/* return the entries of alist sorted non-descending.
*/
```

C. `mode(alist)`

```
/* return the mode of alist by calling
* average-calculator-2.count-by-value(alist),
* and returning a value with the highest count.
*/
```

ii. `average-calculator-2`

A. `count-items(alist)`

```
/* return the number of items in alist.
*/
```

B. `count-by-value(alist)`

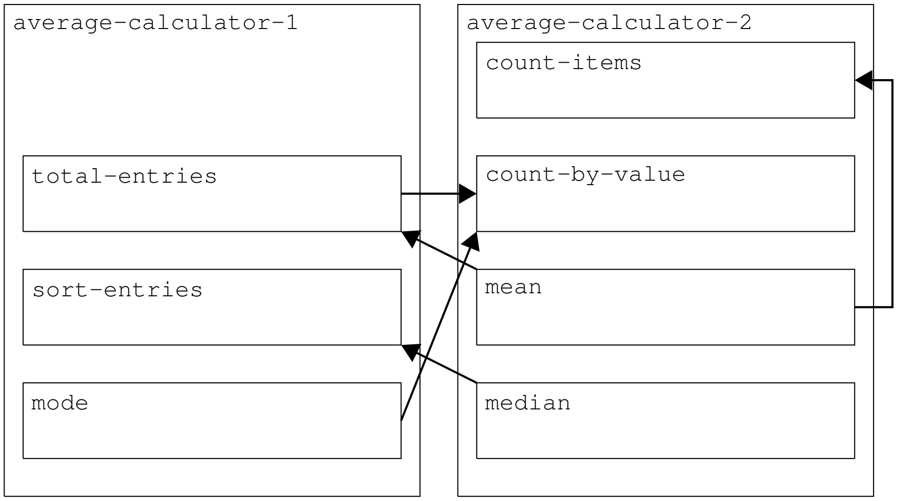
```
/* return a list of pairs (value, count)
*/
```

C. `mean(alist)`

```
/* compute the mean as the quotient
* average-calculator-1.total-entries(alist)
* over average-calculator-2.count-items(alist)
*/
```

D. `median(alist)`

```
/* compute the median as the middle element
* or mean of the the middle two elements of
* average-calculator-1.sort-entries(alist)
*/
```



[4] 4. Give an example of a pair of modules, `module-a` and `module-b`, such that

i. each of `module-a` and `module-b` has **low** cohesion, and

ii. the pair `module-a` and `module-b` has **loose** coupling.

Briefly explain why your example is correct. Note, your description does not need to be highly detailed; describing the functionalities of `module-a` and `module-b` in broad strokes will suffice.