CISC/CMPE 223 - Assignment 5 (Winter 2022)
Due: Thursday April 7, 2:00 PM

Regulations on assignments

- The assignments are graded according to the correctness, preciseness and legibility of the solutions. All handwritten parts, including figures, should be clear and legible. This assignment is marked out of 20 possible marks.
- Please submit your solution in onQ before the due time. The submission must be in one of formats: .PDF, .JPG, .PNG, .DOCX.
- **The assignment must be based on individual work.** Copying solutions from other students is a violation of academic integrity. See the course onQ site for more information.

1. Use the array-component assignment axiom (two times in case (b)) to find the weakest sufficient pre-condition `P` for the following code fragments:

   (a) (1 mark)

   ```
   ASSERT( P ) /* determine what is P */
   A[j] = A[i];
   ASSERT( A[k] >= A[j] )
   ```

   (b) (3 marks)

   ```
   ASSERT( P ) /* determine what is P */
   A[j] = x;
   A[i] = A[k];
   ASSERT( A[j] == 2 )
   ```

   Above `x` is an integer variable, `A` is an array of integers and we assume that all the subscripts are within the range of subscripts for `A`.

   In both cases, write the assertion `P` first using the notation from the array-component assignment axiom, and then rewrite `P` in a **logically equivalent and simplified form** that does not contain any notation $(A \mid I \mapsto E)$ (as in examples on pp. 85–86 and in examples covered in class).

2. (6 marks) Assume a declarative interface where `n` and `max` are constant integers. Also `A` is an array of integers and we know that the entries in the segment `A[0:max]` are defined. Consider the following (partial) correctness statement:

```
ASSERT( 1 <= n < max )
{ int i; i = 1;
  A[0] = 2;
  while( i < n ) { A[i] = A[i-1] + 2*i;
                   i = i+1;
                 } //end while
}
ASSERT( ForAll(k = 0; k < n) A[k] == k*k + k + 2 )
```

Choose a loop invariant and give a **complete proof tableau** by adding all the intermediate assertions. Clearly state any mathematical facts used.

Also make an argument for termination by including a suitable assertion in the loop invariant.

3. (2 marks) Show that the following inference rule for correctness statements is not generally valid:

$$\frac{P_1\{C\}Q_1 \qquad P_2\{C\}Q_2}{P_1 \mid\mid P_2\{C\}Q_1 \&\& Q_2}$$

Hint. In order to show that the inference rule is not valid (i.e., it is unsound), you need to give an example of valid premises such that the conclusion is invalid.

4. (3 marks) For an integer z, power$(2, z)$ denotes "two to the power z", (that is, 2^z).

Is the following specification implementable? Justify your answer!

```
bool SizeOfUniverse(int z);
/* Returns true if power(2, z) is  greater than
   the maximum number of atoms in the universe any time
   in the past or future; otherwise returns false.
*/
```

Note: If the number of atoms continues to grow indefinitely with time, the maximum number may be unbounded. In this case `SizeOfUniverse` should return false for any integer `z` given as input.

5. (5 marks) Is the following specification implementable? Justify your answer.

```
bool HaltsOnSelf(FILE *func);
/* Returns true if func contains the definition of an int
   function with one FILE parameter, and that function
   terminates if applied to a file containing its own
   definition; otherwise HaltsOnSelf returns false.
*/
```

Note: Question 5 is Exercise 12.3 on page 249 in the text. As mentioned in the textbook, the argument is similar to the one used for the specification of `halts`. However, you are expected to give a detailed argument, that is, you should write a program analogous to *Program 12.1* (page 248) and then explain why it necessarily results in a contradiction.