Assignment 7
Due on: 2019-09-23, 23:09 IST.

1. Write an optimisation program, the maximum benefit is obtained by optimising functions that are called only once.
   - True
   - False
   - No, the answer is incorrect.

2. Feedback: Functions that are called several times are more likely to take more total time to execute, even if they are not optimised.
   - Accepted
   - Accepted Answer

3. A wrong program will always access the random number faster than a long once.
   - True
   - False
   - No, the answer is incorrect.

4. Feedback: After compilation and optimisation especially when linking it is quite possible to make a program run longer while still speeding the running time.
   - True
   - False
   - No, the answer is incorrect.

5. Castle memory has to be faster than main memory to be of any use.
   - True
   - False
   - No, the answer is incorrect.

6. Feedback: The size of castle memory is in storage temporarily sizes that are accessed often, if it is slower than main memory, then you could just use the main memory and not worry about caching.
   - True
   - False
   - No, the answer is incorrect.

7. Feedback: Function following instructions execute way faster if the same function is executed more times over and over again.
   - True
   - False
   - No, the answer is incorrect.

8. Feedback: Software pipelining can result in slower execution if not applied properly.
   - True
   - False
   - No, the answer is incorrect.

9. For the code shown below, numbers on the rows are true numbers. Answer the following questions in markdown:

   ```c
   1 // function declaration
   2 float magnitude(float a, float b) {
   3     return sqrt(a*a + b*b);
   4 }
   5 // loop
   6 for (i=1 ; i<5 ; i++) {
   7     y[i][i] = x[i][0] + cos(x[i][i+1]) - x[1][i] + sin(x[3][i+1]) + sin(x[4][i+1]) + cos(x[3][i+1]);
   8     y[2][i] = x[i][0] + sin(x[i][i+1]) + sin(x[i][i+1]) + cos(x[i+1][i+1]);
   9     mg[i] = magnitude(y[2][i], y[2][i+1]);
   10 }
   ```

   10. What effect will function inlining have on the size of the code?
   - Increase
   - Decrease
   - Rename the name
   - No, the answer is incorrect.

   11. Feedback: The last effect is the same, the function is called only once, and is just one line of code. So inlining will actually reduce code size.
   - Accepted
   - Accepted Answer

   12. There is a write-after-read (WAX) dependency between lines 7 and 6.
   - True
   - False
   - No, the answer is incorrect.

   13. Feedback: They do not rely on a value used by each other.
   - Accepted
   - Accepted Answer

   14. There is a read-after-write (RAW) dependency between lines 7 and 6.
   - True
   - False
   - No, the answer is incorrect.

   15. Feedback: RAW is not a dependency; there can happen in any order without causing issues, so there is no dependency, so there is no dependency on the order in which they read happen.
   - Accepted
   - Accepted Answer

   16. There is a read-after-read (RAX) dependency between lines 6 and 8.
   - True
   - False
   - No, the answer is incorrect.

   17. Feedback: Since the dependency does not depend on the output of the loop, it should not need the value of y[2][i+1] only after line 8 has been executed.
   - Accepted
   - Accepted Answer

   18. Which of the following optimisations will help to make the code run faster?
   - Common subexpression elimination
   - Dependence elimination
   - Control hoisting
   - No, the answer is incorrect.

   19. Feedback: The optimisations will have a positive effect on this code.
   - Accepted
   - Accepted Answer

   20. Control hoisting
   - Common subexpression elimination
   - Dependence elimination
   - No, the answer is incorrect.