Unit 9 - Week 7

Assignment 7

The due date for submitting this assignment has passed.
As per our records you have not submitted this assignment.

Due on 2019-09-25, 23:59 IST.

1) While optimizing programs, the maximum benefit is obtained by optimizing functions that are called only once — *1 point*

- True
- False

No, the answer is incorrect.
Score: 0
Feedback:
Solution: *Functions that are called several times are more likely to take more total time to execute,even if they are individually faster*
Accepted Answers:
False

2) A shorter program will always execute faster than a long one — *1 point*

- True
- False

No, the answer is incorrect.
Score: 0
Feedback:
Solution: *After compilation and optimization (especially unrolling etc) it is quite possible to make a program longer while still reducing the running time*
Accepted Answers:
False

3) Cache memory has to be faster than main memory to be of any use — *1 point*

- True
- False

No, the answer is incorrect.
Score: 0
Feedback:
Solution: *The idea of cache memory is to temporarily store values that are accessed often. If it is slower than main memory, then you could just use the main memory and not worry about caching*
Accepted Answers:
True

4) Function inlining can increase code size but still result in faster code — *1 point*

- True
- False

No, the answer is incorrect.
Score: 0
Feedback:
Solution: *Function inlining can increase code size if the same function is used more than once and inlined in both places. But it may still be faster because overhead of function call is eliminated*
Accepted Answers:
True

5) Software pipelining can result in slower execution if not applied properly — *1 point*

- True
- False

No, the answer is incorrect.
Score: 0
Feedback:
Solution: *Well, that is true of everything. In the case of software pipelining, there is a prologue and epilogue that can make the program longer and slower if there is not enough work done in the loop portion*
Accepted Answers:
True

For the code shown below, numbers on the left are line numbers. Answer the following questions 6-10 :

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

6) What effect will function inlining have on the size of the code? — *1 point*

- increase
- decrease
- remain the same

No, the answer is incorrect.
Score: 0
Feedback:
Solution: *In this case, the function is called only once, and is just one line of code. So inlining will actually reduce code size*
Accepted Answers:
decrease

7) There is a write-after-read (WAR) dependency between lines 7 and 8 — *1 point*

- True
- False

No, the answer is incorrect.
Score: 0
Feedback:
Solution: *They do not modify a value used by each other.*
Accepted Answers:
False

8) There is a read-after-read (RAR) dependency between lines 7 and 8 — *1 point*

- True
- False

No, the answer is incorrect.
Score: 0
Feedback:
Solution: *RAR is not a dependency - these can happen in any order without causing issues,so there is no restriction or dependency on the order in which they must happen*
Accepted Answers:
False

9) There is a read-after-write (RAW) dependency between lines 8 and 9 — *1 point*

- True
- False

No, the answer is incorrect.
Score: 0
Feedback:
Solution: *Line 9 depends on the output of line 8, so it should read the value of y[2*i+1] only after line 8 has written to it*
Accepted Answers:
True

10) Which of the following optimizations will help to make the code run faster? — *1 point*

- Constant folding
- Common subexpression elimination
- Code hoisting

No, the answer is incorrect.
Score: 0
Feedback:
Solution: *All the optimizations will have a positive effect on this code*
Accepted Answers:
Constant folding
Common subexpression elimination
Code hoisting