CS 4400: Computer Systems

Midterm Exam 2 SAMPLE SOLUTIONS
Fall 2010

1. (a) Function `factorial_u2` will return 0 whenever `n` is odd.
   (b) Change loop test to `i > 1`. (Note that no "clean-up" is needed after the loop.)
   (c) Performance is limited by the 4-cycle latency of integer multiplication.
   (d) The multiplication `z = i * (i-1)` can overlap with the multiplication `result * z` from the previous iteration (instead of occurring sequentially, as in `factorial_u2`). Thus, each iteration requires 4 cycles and processes two elements, giving a CPE of 2.0.
   (e) Speedup due to `gcd`: $\frac{1}{(1-a)+a/k} = \frac{1}{.45+.55/11} = \frac{1}{.5} = 2$
   Speedup due to `average`: $\frac{1}{(1-a)+a/k} = \frac{1}{.59+.41/41} = \frac{1}{.6} = 1\ 2/3$
   `gcd` results in the largest speedup.

2. (a)

11	10	9	8	7	6	5	4	3	2	1	0
CT	CT	CT	CT	CT	CT	CT	CT	CI	CI	CI	CO

   (b)

11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	0	1	1	0	1	0	0

   (c)

Parameter	Value
Cache Offset (CO)	0x0
Cache Index (CI)	0x2
Cache Tag (CT)	0xE3
Cache Hit? (Y/N)	N
Cache Byte Returned	0x—

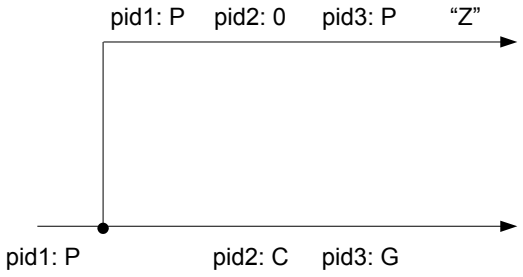
3. (a) 128 misses in the first loop, 128 misses in the second loop
   256 misses / 1024 references = 25% miss rate
   (b) 128 misses / 1024 references = 12.5% miss rate
   (c) 256 misses / 1024 references = 25% miss rate

4. (a) `REF(d.1)` → `DEF(d.1)`
   `REF(d.2)` → `DEF(d.2)`
   (b) `REF(ptr.1)` → (ERROR)
   `REF(ptr.2)` → (ERROR)
   (c) `REF(foo.1)` → `DEF(foo.2)`
   `REF(foo.2)` → `DEF(foo.2)`

5.

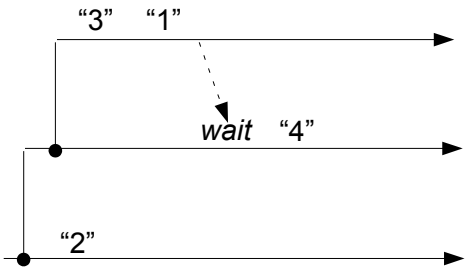
Symbol	*Entry*	*Type*	*Module*	*Section*
`find_max`	yes	extern	`max.o`	`.text`
`input_a`	yes	local	`main.o`	`.data`
`input_b`	no	–	–	–
`max`	yes	extern	`max.o`	`.bss`
`rand`	yes	extern	`libc.a`	`.text`
`test_find_max`	yes	local	`main.o`	`.text`

6.



Only possible output: `Z`

7. (a)



(b) `2314`, `3214`, `3124`, `3142`

8. The bug is that signals will be received before the parent and child set up their signal handlers. A good solution would be to block `SIGUSR1` before the fork and then both processes unblock it after setting their signal handlers.