CSC 209H1 S 2015 Midterm Test
Duration — 50 minutes
Aids allowed: none

Student Number: ____________________

Last Name: ____________________ **First Name:** ____________________

Lecture Section: L0201 **Instructor: Simion**

*Do **not** turn this page until you have received the signal to start.*
(Please fill out the identification section above, **write your name on the back of the test**, and read the instructions below.)
Good Luck!

This midterm consists of 5 questions on 7 pages (including this one). *When you receive the signal to start, please make sure that your copy is complete.* Comments are not required, although they may help us mark your answers. They may also get you part marks if you can't figure out how to write the code. Answers that contain both correct and incorrect or irrelevant statements will not get full marks.
If you use any space for rough work, indicate clearly what you want marked.

1: ______/ 5
2: ______/ 3
3: ______/ 3
4: ______/ 3
# 5: ______/13

TOTAL: ______/27

SOLUTIONS

LEC 0201

Question 1. [5 MARKS]

Part (a) [1 MARK]

If your current working directory is `/home/user` give 2 different ways to run a program called `markone` that is stored in `/home/user/bin`.

SOLUTION:

```
/home/user/bin/markone
bin/markone ./bin/markone
```

Part (b) [3 MARKS]

A user types the following at the command line:

```
ant bug cat < dog > eel
```

The program or programs to be executed are: ant

The arguments to the program or programs are: bug cat

For each of the processes created, standard input comes from: dog

For each of the processes created, standard output goes to: eel

For each of the processes created, standard error goes to: console

Part (c) [1 MARK]

Write a line of code that divides integer variable `a` by 2 without using the division operator (/).

```
a = a >> 1;
```

Question 2. [3 marks]

Circle the correct answer, and briefly explain it.

[✓ TRUE] FALSE It is possible that a parent process exits before its child process finishes.

Explain:

TRUE [✓ FALSE] In a parent process a wait call always blocks.

Explain:

If the child process has already terminates (or was never created) then the wait call will not block.

TRUE [✓ FALSE] A process running in the background can't be killed.

Explain:

Question 3. [3 marks]

Suppose we write a Makefile for assignment 1 with the following rules:

```
addecho : addecho.c
    gcc -Wall -g -o addecho addecho.c
remvocals : remvocals.c
    gcc -Wall -g -o remvocals remvocals.c
```

Part (a) [1 mark] Write a rule so that when we run `make` both programs are recompiled if necessary.

```
all :  addecho remvocals
```

Part (b) [2 marks] Write a rule with the target `test` that will ensure that `remvocals` is compiled if it is out of date, and will run `remvocals` with the arguments `simple.wav testout.wav`

```
    test : remvocals
        remvocals simple.wave testout.wav
```

Question 4. [3 MARKS]

Consider the following program.

```
int main() {

    int r = fork();

    if(r > 0) {
        int status;
        fprintf(stderr, "A");

        if(wait(&status) != -1) {
            if(WIFEXITED(status)) {
                fprintf(stderr, "%d", WEXITSTATUS(status));
            }
        }
        fprintf(stderr, "B");

    } else {
        fprintf(stderr, "C");
    }
    fprintf(stderr, "D");
    return 4;
}
```

Check all of the boxes that are valid output for the above program.

□ A4BCDD

☒ CAD4BD

□ BA4CDD

□ AC4BDD

☒ ACD4BD

□ ADC4BD

Question 5. [13 MARKS]

Given the following struct definition:

```
struct entry {
    char word[16];
    char def[64];
};
```

Part (a) [6 MARKS] Complete the code below so that it is correct, and the output is:

```
quokka : small macropod
macropod : marsupial family that includes kangaroos
```

```
// Store newword and newdef in the fields of e

void set_entry(______________________________ e, char *newword, char *newdef) {






}

void print_entry(________________________________ e) {


    printf("%s : %s\n", __________________________, ____________________);
}

int main() {

    struct entry *dict = _________________________________________________;


    set_entry(_____________________________, "quokka", "small macropod");


    set_entry(_____________________________, "macropod",
                                 "marsupial family that includes kangaroos");


    print_entry(__________________);


    print_entry(__________________);

}
```

```
void set_entry(struct entry *e, char *newword, char *newdef) {
    strncpy(e->word, newword, 16);
    strncpy(e->def, newdef, 64);
}

void print_entry(struct entry e) {
    printf("%s : %s\n", e.word, e.def);
}

int main() {
    struct entry *dict = malloc(10 * sizeof(struct entry));

    set_entry(&d[0], "quokka", "small macropod");
    set_entry(&d[1], "macropod", "marsupial family that includes kangaroos");

    print_entry(d[0]);
    print_entry(d[1]);
}
```

Part (b) [1 MARK] How many bytes are required to store the following two variables.

`struct entry *dict:` 8 (or 4) ____________________ `struct entry e:` 80 bytes ____________________

Part (c) [6 MARKS]

Give the declaration for the variable to make each of the following statements correct, or if there is an error in the statements write "ERROR".

```
    struct entry *dict;
    struct entry words[5];
```

	Declaration of x
`x = &dict;`	`struct entry **x;`
`x = dict->word`	`char *x;`
`x = words[1]`	`struct entry x;`
`x = &words[1]`	`struct entry *x;`
`x = words[1].word[2]`	`char x`
`x = dict[0].word`	`char *x`