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

Student Number: |_|_|_|_|_|_|_|_|_|_|

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

Instructor: Craig

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 8 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.
No error checking is required unless you are specifically requested to do it for an individual question.
You do not need to provide the include statements for your programs.
If you use any space for rough work, indicate clearly what you want marked.

# 1: ______/10
2: ______/ 5
3: ______/ 4
4: ______/ 5
5: ______/ 6

TOTAL: ______/30

Question 1. [10 marks]

Complete the code below according to the comments.

```
typedef struct month{
   char name[10];
   int num_days;
   int num_holidays;
   char **holidays;
} Month;

/* Return a value greater than 0 if m1 is longer than m2, less than 0 if m2
 * is longer than m1 and 0 if they have the same number of days */
int longer(Month *m1, Month *m2) {
    return m1->num_days - m2->num_days;
}

int main(int argc, char **argv) {

    struct month dec;

    // Set the number of days in dec to 31.




    // Set the name of dec to the value of the first command-line argument.
    // If the value provided by the user is too long, truncate it.








    // Allocate space for an array of 2 december holidays
    // and set the holidays to mutable strings "christmas" and "hanukkah".
    // Change the first letter of each of the holidays to uppercase so they are
    // "Christmas" and "Hanukkah". Do this by changing the strings in place,
    // rather than by assigning a completely new string.



    Month *june;

    // Create a second month pointed to by the variable june declared above.



    // Set the number of days in june to 30.



    // Call longer correctly on june and dec so this code works.


    if (                                                                   ) {
        printf("December is longer than June.\n");
    }
    return 0;
}
```

Question 2. [5 MARKS]

In the table below, provide the value of `cars` and of `*p` **after** the corresponding code fragment has finished executing. If either value is not yet defined, write `UNDEFINED` and if the code would not execute because of an error, write `ERROR`.

Code Fragment	`cars`	`*p`

`int *p;`
`int cars = 145;`
`p = &cars;`
`*p = 3;`		

`int cars = 145;`
`int *p = &cars;`
`cars = 77;`		

`int trucks = 6;`
`int *p = &trucks;`
`int cars = *p + 5;`		

`int trucks = 4;`
`int cars = 8;`
`int *p;`
`p = &trucks;`
`*p = 6;`
`p = &cars;`		

`int cars = 2;`
`int *p = &cars;`
`int **q = &p;`
`int trucks = 3;`
`*q = &trucks;`		

Question 3. [4 MARKS]

Part (a) [2 MARKS]

Write a shell command that given a shell variable `FILENAME` that holds the name of a non-existing file, creates a file by this new name that contains only the line `hello world`. Do not write a C program to solve this problem.

Part (b) [2 MARKS]

Give the command to reset the permissions of file `myfile` so that anyone can read the file, but only the owner and members of the group can write it. Nobody should be able to execute the file.

Question 4. [5 MARKS]

Consider the following code fragment as part of the main function in a C program.

```
char *top[2];
top[0] = malloc(10);
strcpy(top[0], "stuff");
top[1] = "more stuff";
```

Part (a) [3 MARKS]

Draw a picture of the memory that has been allocated or assigned. Label each piece of allocated memory as “stack”, “heap” or “read-only”. When one memory location holds the address of another, you may show this with an arrow (or you may make up a memory address).

Part (b) [1 MARK]

Write a line of C code that if executed following the above fragment, would result in a dangling pointer.

Part (c) [1 MARK]

Write a line of C code that if executed following the above fragment, would result in a memory leak.

Question 5. [6 MARKS]

You have a text file that on the first line has two integers separated by white space. Below you will write two different programs that accomplish roughly the same thing. Some of the lines of code will be the same but there are important differences in the program descriptions. Follow the instructions carefully. When you need to make assumptions, go ahead and state them clearly. You do not need to write error checking for your function calls.

Part (a) [4 MARKS]

Write a C program that reads the file's **name** from standard input, opens that file, reads the two integers and prints their sum to standard out.

Part (b) [2 MARKS]

Write a program that reads the **contents** of the first line of that same file from standard input and prints the sum of the two integers to standard out. In other words your program could be compiled into the executable `a.out` and then called on the file `ages.input` using the command `cat ages.input | ./a.out`

[Use the space below for rough work. This page will not be marked unless you clearly indicate the part of your work that you want us to mark.]

C function prototypes:

```
int fclose(FILE *stream)
FILE *fopen(const char *file, const char *mode)
size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);
void free(void *ptr)
int fscanf(FILE *restrict stream, const char *restrict format, ...);
int fseek(FILE *stream, long offset, int whence)
size_t fwrite(const void *ptr, size_t size, size_t nmemb, FILE *stream);
void *malloc(size_t size);
void perror(const char *s)
int scanf(const char *restrict format, ...);
char *strchr(const char *s, int c)
size_t strlen(const char *s)
char *strncat(char *dest, const char *src, size_t n)
int strncmp(const char *s1, const char *s2, size_t n)
char *strncpy(char *dest, const char *src, size_t n)
char *strrchr(const char *s, int c)
```

Excerpt from scanf man page:

```
RETURN VALUES
     scanf returns the number of input items assigned.  This can be
     fewer than provided for, or even zero, in the event of a matching fail-
     ure. The value EOF is returned if an input failure occurs before any
     conversion such as an end- of-file occurs.
```

Excerpt from strcpy/strncpy man page:

```
The stpcpy() and strcpy() functions copy the string src to dst (including
the terminating ‘\0’ character).

The stpncpy() and strncpy() functions copy at most n characters from src
into dst.  If src is less than n characters long, the remainder of dst is
filled with ‘\0’ characters.  Otherwise, dst is not terminated.
```

Print your name in this box.