= COMP1917 Final Exam : 2013s1 =

== Time allowed: 150 minutes ==
== Total number of questions: 20 ==
== Total number of marks: 85 ==

There is one mark for following the examination instructions.

Your answers can be submitted by pressing save on this application.

You may submit your solutions as many times as you like. ONLY the last submission will be marked.

Write your name and stusent number on the top of each sheet of rough working paper you use, this will not be marked but we may inspect it in borderline cases. You will be provided with an electronic copy of the textbook.

You must hand in ALL writing paper at the end of the exam.

Once the exam has commenced you may not leave until the end of the exam period (morning session only).

Calculators may not be used. Compilers or other software tools may not be used. You may only use the specified editor and pdf viewer. The eating of toast in the exam room is prohibited. You must not attempt to communicate with any other person, or access other computers or any internet resources. If you do not follow these instructions you will get zero marks for the exam and a possible charge of academic misconduct.

Unless advised otherwise you may not use any language features or library functions not covered in class. Code which does not comply with the course style guide, which contains a security vulnerability, or which is overly complex or unclear will be penalised.

Questions (and sub-questions) are not worth equal marks. Answer all questions.

== PART A: Short Answer Questions ==

Your working is not marked in Part A. Questions are not all of equal value. Keep your answers clear and very brief. Messy or long answers will not be marked.

=== Question 0 ===

(2 Marks)

Consider the following C definition:

`char str[] = "Hello World!";`

The size of the array str in bytes on a 32 bit lab machine would be:

.[A] 4
.[B] 12
.[C] 16
.[D] 32
.[E] 13

=== Question 1 ===

(2 Marks)

```
{{{
   int i = G;
   while (i > 1) {
      printf("Hello world!\n");
      i--;
   }
}}}
```

What value should G be given so that the string "Hello world!\n" is printed exactly 10 times?

 .[A] 10
 .[B] 11
 .[C] 12
 .[D] 13
 .[E] None of the above

=== Question 2 ===

(2 Marks)

Consider the following C function prototype:

`int f(int *x, int y);`

Which of the following statements is true of argument `x`?

 .[A] it is passed by magic
 .[B] It is passed by copying
 .[C] It is passed by reference
 .[D] It is passed by abstraction
 .[E] None of the above

=== Question 3 ===

(2 Marks)

Given the following fragment of code

```
{{{
char str1[] = "hello";
char str2[] = "hello";

if (str1 == str2) {
   printf ("equal\n");
} else {
   printf ("not equal\n");
}
}}}
```

What will this program print out?

_____
_____

Explain why.

_____
_____

=== Question 4 ===

(3 Marks)

The purpose of the following three lines of code is to swap the values in variables a and b:

```
{{{
c=a;
a=b;
b=c;
}}}
```

The three lines of code below are the same as the lines above, but in a different order:

```
{{{
a=b;
b=c;
c=a;
}}}
```

In one sentence in the box below, describe the purpose of the second block of three lines.

(NOTE: Tell us what the second block of three lines of code do all by themselves. Do NOT think of those second three lines as being executed after the first three lines of code.)

_____
_____

=== Question 5 ===

(3 Marks)

Consider the following code fragment:

```
{{{
   int x[] = {0, 1, 2, 3};
   int temp;
   int i = 0;
   int j = (sizeof (x)/sizeof (int)) - 1;
   while (i < j) {
      temp = x[i];
      x[i] = x[j];
      x[j] = 2*temp;
      i++;
      j--;
   }
}}}
```

After this code is executed, array x contains the values:

.[A] {3, 2, 2, 0}
.[B] {0, 1, 2, 3}
.[C] {3, 2, 1, 0}
.[D] {0, 2, 4, 6}
.[E] {6, 4, 2, 0}

=== Question 6 ===

(3 Marks)

Assume the array
{{{
int numbers[SIZE];
}}}
has been previously declared and given values.

In plain English, explain what the following segment of code does:
{{{
```
   int valid = TRUE;
   int i = 0;
   while (i < (SIZE-1)) {
      if (numbers[i] > numbers[i+1]) {
         valid = FALSE;
      }
      i++;
   }
```
}}}

_____
_____

=== Question 7 ===

(3 Marks)

Below are the hexadecimal contents of memory locations F000..F00D on a 32 bit little endian computer currently running a program:

{{{
```
 F000  22
 F001  6E
 F002  6F
 F003  6D
 F004  20
 F005  54
 F006  6F
 F007  61
 F008  73
 F009  74
 F00A  21
 F00B  00
 F00C  82
 F00C  F0
```
}}}

The currently running program was written in C and includes the following three (consecutive) lines of printfs involving an int m
{{{
```
printf ("%p\n", &m);
printf ("In hex is %x\n", m);
printf ("In decimal is %d\n", m);
```
}}}

The first printf prints out 0xf005, in the space below write what ONE of the other two printfs would print (you may select whichever you wish).

_____

=== Question 8 ===

(3 Marks)

This question relates to the same computer and program (and memory contents)
as the previous question.

The currently running program was written in C and includes the following
two (consecutive) lines of printfs
{{{
```
printf ("%p\n", b);
printf ("%s\n", b);
```
}}}

The first printf prints out 0xf005, in the space below write what the other
printf would print.

_____

=== Question 9 ===

(4 Marks)

When the following code is compiled and executed it eventually executes
the printf and terminates. What value does it print?

{{{
```
#include <stdio.h>
#define NINE 9

int nine (int n);
int nein (int n);
int main (int argc, char *argv[]) {

   int Nine = 3;
   printf ("%d\n", nine (Nine));
   return !NINE;
}

int nein (int nein) {
   return ( !(nein == ((NINE*nein) % NINE)) );
}

int nine (int Nein) {
   int Nine = NINE;
   while (nein (Nine)) {
      Nine += Nein;
      Nine = Nine % NINE;
   }
   return Nine;
}
```
}}}

_____

=== Question 10 ===

(3 Marks)

{{{
```
// helloWorld.c
```

```
#include <stdio.h>
#include <stdlib.h>

int main (int argc, char *argv[]) {
   char hello[5];
   hello [0] = 'H';
   hello [1] = 'e';
   hello [2] = 'l';
   hello [3] = 'l';
   hello [4] = 'o';

   char world[5];
   world [0] = 'W';
   world [1] = 'o';
   world [2] = 'r';
   world [3] = 'l';
   world [4] = 'd';

   printf ("%s %s!\n", hello, world);
   return EXIT_SUCCESS;
}
```

When a programmer ran the helloWorld.c program above on their laptop they got the following output:

```
Hello WorldHello!
```

Briefly explain the most likely reason the program produced this output rather than simply printing ``Hello World!''.

_____
_____
_____
_____

=== Question 11 ===

(4 Marks)

Given the following code:

```
int q;
int *p;
q = 24;
*p = 42;
q = *p;
printf("The value of q is %d\n",q);
```

What is the main problem you would raise in reviewing the above code?

_____
_____
_____

What result would you expect from compiling/running the code and why?

_____
_____

_____

=== Question 12 ===

(4 Marks)

Given the following code:

```
{{{
char str[] = "a dog!";
int len = strlen(str);
str[len] = '\n';
str[len+1] = '\0';
printf("%s\n",str);
}}}
```

What is the main problem you would raise in reviewing the above code?

_____
_____
_____

What result would you expect from compiling/running the code and why?

_____
_____
_____

== PART B: Programming I ==

Make your answers as clear and easy to understand as possible. Provide brief comments in your code where necessary. Confusing or illegible solutions will lose marks.

In this part if you do not wish your answer for a question to be marked, write 1 SYMPATHY MARK PLEASE in the space for the answer. If you do this you will be awarded one sympathy mark and your answer for that question will not be marked.

=== Question 13 ===

(5 Marks)

Write an 8008 program to sum together (mod 256) the numbers stored in memory cells 200..250 inclusive and print out the answer (the answer should be the only thing printed).

For example if the cells all contained 1: your program should print 51.

For example if the cells all contained 10: your program should print 254. (as the sum is 510, and 510 mod 256 is 254.

Include C-style // comments in your machine code. Make your answer clear with lot of comments and a line by line explanation of what you are doing. We'll execute your code and award full marks if it works or partial marks otherwise.

_____
_____
_____
_____
_____

_____
_____
_____
_____
_____
_____
_____

=== Question 14 ===

(6 Marks)

Consider the following code fragment:

```
{{{
void doCalculation (double x, int y);

// This program reads in two numbers, a double and an int.
// It prints out the first number divided by 2
// and the second number incremented by 10
// It does not perform any error checking.
int main (int argc, char * argv[]){
    double x;
    int y;

    scanf("%lf %d",&x,&y);
    doCalculation(x,y);
    printf("%lf %d\n",x,y);

    return EXIT_SUCCESS;
}

void doCalculation (double x, int y){
    x = x/2;
    y = y + 10;
}
}}}
```

.(a) Explain what is wrong with this program and why it does not work as expected from the comment.

_____
_____
_____
_____

.(b) Write a different prototype for the function doCalculation to successfully change the values in main.

_____
_____
_____
_____

.(c) Implement this different doCalculation (ie write the code for it).

_____
_____
_____
_____

.(d) How would you now call the function within the main function?

_____
_____
_____
_____

=== Question 15 ===

(5 Marks)

Write out the contents of a small file twos.c which complies with the course style guide and, when compiled and executed on a lab machine with the commands:

```
{{{
gcc -Wall -Werror -O -std=c99 -o twos twos.c
./twos
}}}
```

reads in a number n and prints out the number of times it can be halved before obtaining an odd number.

For example given 5 it would print 0 (as 5 is itself odd), and given 12 it would print 2 (as half of twelve is six, and half of six is three, which is odd, hence 12 can be halved twice.)

```
{{{
$ ./twos
5
0
$ ./twos
6
1
$ ./twos
12
2
$ ./twos
64
6
}}}
```

You may assume that n is a positive integer between 1 and 1,000,000.

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

=== Question 16 ===

(5 Marks)

The following function is to print out its three distinct input parameters in ascending order. The function is to perform this sorting using nested if statements only, NOT using arrays or while loops. Write the code fragment to go in the section marked YOUR CODE HERE. Assume Julian will be writing the code in the section marked JULIAN TO DO so you do not write that part.

{{{
```
void threeBears (int a, int b, int c) {
   int daddy, mummy, baby;

   if (a<b) {
      // YOUR CODE HERE
   } else {
      // JULIAN TO DO
   }

   printf ("The biggest is %d\n", daddy);
   printf ("The middle is %d\n", mummy);
   printf ("The smallest is %d\n", baby);

}
```
}}}

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

== PART C: Programming II ==

Make your answers as clear and easy to understand as possible. Provide brief comments in your code where necessary. Confusing or illegible solutions will lose marks.

In this part if you do not wish your answer for a question to be marked, write 1 SYMPATHY MARK PLEASE in the space for the answer. If you do this you will be awarded one sympathy mark and your answer for that question will not be marked.

=== Question 17 ===

(7 Marks)

Write a function which implements the prototype

{{{
```
void classify (int array[], int n);
```
}}}

which inspects an array of n elements to find which one of the following cases is true and prints a single letter identifying the case.

a. array contains no zeros.

b. array contains only one zero.

c. the first two zeros in the array are separated by an even number of non-zeros.

d. the first two zeros in the array are separated by an odd number of non-zeros.

The function should either print "a", "b", "c", or "d".

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

=== Question 18 ===

(12 Marks)

You are to design an ADT to represent a complex number. In this question you will write the interface header file Complex.h, the file Complex.c which implements the ADT interface functions, and a file ComplexUser.c which uses the ADT to perform a simple computation.

The ADT needs to have an interface function:

Complex multiply (Complex a, Complex b)

which returns the product of two complex numbers, and an interface function:

Complex add (Complex a, Complex b)

which returns the sum of two complex numbers. It may also include other interface and helper functions if you wish.

Write the file Complex.h:

_____
_____
_____
_____
_____
_____
_____

Write the file Complex.c:

_____

_____
_____
_____
_____
_____
_____
_____
_____

Write a file ComplexUser.c which hash includes Complex.h and uses the
ADT to compute (0.5 + 0.5i) * (0.5 + 0.5i) + (0.1 + 0.9i) and print out
the answer:

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

== PART D: Challenge Part ==

Partial solutions for this question (i.e. attempts worth less than 50%)
will score no marks in this part.

If you do not wish your answer for the question to be marked, write 1 SYMPATHY
MARK PLEASE in the space for the answer. If you do this you will be awarded
one sympathy mark and your answer for the question will not be marked.

=== Question 19 ===

(6 Marks)

Write an 8008 function which is passed the address of the first character
of a C-style string and which prints out the string.

Assume the frame for the function starts at memory cell 200, and your
function starts at cell 100.

Make sure your code does implement a FUNCTION, eg it must return at the
end save and restore registers and so forth. In your answer do not
implement the main function, just show the fragment of memory starting
from cell 100 which implements your function.

We'll execute your code and award full marks if it works or zero marks otherwise. You
may (and should) include // c style comments if you wish
for your own debugging and style purposes, these will be safely stripped
out before we execute your code.

_____
_____
_____
_____