UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

APS 105 — Computer Fundamentals
Midterm Examination
February 28, 2017
1:15 p.m. – 3:00 p.m.
(105 minutes)

Examiners: B. Li, J. Rose, M. Stumm

Exam Type A: This is a "closed book" examination; no aids are permitted.

Calculator Type 4: No calculators or other electronic devices are allowed.

All questions are to be answered on the examination paper. If the space provided for a question is insufficient, you may use the last page to complete your answer. If you use the last page, please direct the marker to that page and indicate clearly on that page which question(s) you are answering there.

You must use the C programming language to answer programming questions. You are not required write `#include` directives in your solutions. Except those excluded by specific questions, you may use functions from the `math` library as necessary. If you are unsure about the meaning of a question, make an assumption and write that assumption down as part of your answer.

The examination has 21 pages, including this one.

Circle your lab section — (**1 mark will be deducted** if you fail to indicate your lab section):

PRA0101	**PRA0102**	**PRA0103**	**PRA0104**
Friday	Friday	Friday	Monday
12-2pm	9-11am	9-11am	12-2pm

First Name: _____ Last Name: _____

Student Number: _____

MARKS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	Total
/2	/2	/2	/2	/2	/2	/2	/2	/2	/6	/6	/6	/7	/7	/50

Question 1 [2 Marks]

Write a single line C statement that computes the expression below for the variable `y` which is computed based on variables `x` and `z`. You can assume that all of the functions in the C `math` library are available, and that `x`, `y` and `z` have already been declared as `double`-type variables.

$$y = \sqrt{\frac{50.0}{1.0 - e^{-x/z}}}$$

Please write your answer here:

Question 2 [2 Marks]

Assume that three integer variables, `a`, `b` and `c` have been declared and set to some value greater than zero. It is possible to compute, for each variable, the remainder when it is divided by 10. Write a single C statement that computes the real-number average of those three remainders.

Please write your answer here:

Question 3 [2 Marks]

Consider the portion of a C program shown below. Assume that variables `a`, `b`, and `c` (which are declared at the top of the program) have their values set as given in each line of the table below, after which the code is executed. For each such case, give the value of the variable `m` in the table *after* the code is executed. Please write your answer directly into the table, under the column labelled `m`.

```
int a, b, c, m;

// assume that the values of a, b, c are set as in the table below
if ((a < b) && (a < c))
  m = a;
else if ((b < a) && (b < c))
  m = b;
else if ((c < a) && (c < b))
  m = c;
else
  m = -1;
```

a	b	c	m
50	90	30	
50	50	50	
70	0	600	
3	2	1	

Question 4 [2 Marks]

Consider the following code segment containing three `if` statements:

```
if (a == b)
  if (c != d)
    if (e == f)
      printf("Are you kidding me????\n");
```

Please write a C code segment that behaves exactly like the above code, but that contains only one `if` statement.

Please write your answer here:

Question 5 [2 Marks]

Write a single C statement that declares a `double` variable `randomSelection`, and initializes it with a number that is randomly selected from the following set: 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95.

Please write your answer here:

Question 6 [2 Marks]

Consider the following code, which uses pointers:

```
int a, b, c, d;
int *e, *f;

a = 5;
b = 6;

e = &c;
f = &d;

*e = a + b;
*f = *e + b;

e = &a;
f = &b;

*e = c + d;
*f = a + b;
```

What are the values of the variables `a` and `b` after this code is executed?

Please write your answer here:

Question 7 [2 Marks]

Assume a program has three boolean variables, `a`, `b`, and `c`, which can each have a value of `true` or `false`. In the space provided below, please list *all* combinations of values for `a`, `b`, and `c`, for which the following code segment does **not** print out "`Success!!`".

```
bool a, b, c;

...
...  // other code here...
...

if (!(a && !(b || c)))
  printf( "Success!!" );

...
```

For example, if your answer were:

```
a: false  b: true  c: false
```

then your answer would not be correct, because `!(false && !(true || false))` is `true` and hence "`Success!!`" would be printed.

List **all** combinations for which "`Success!!`" would **not** be printed here:

Question 8 [2 Marks]

(a) Consider the function:

```
void printSomething(int n);
```

and the following `for` loop that calls the function:

```
for (int i = 0; i < n; i++)
  printSomething(i);
```

Write a code segment that has the same effect as the above `for` loop but only uses a `while` loop (without using a `for` loop or a `do-while` loop).

(b) Assume variable `i` has been declared and has a value. Further, assume the following function exists:

```
int someFunctionCall(int n);
```

Then, consider the following `do-while` loop:

```
...
do
  i = someFunctionCAll(i);
while (i >= 0);
```

Write a code segment that has the same effect as the above code but only uses a regular `while` loop (without using a `for` loop or a `do-while` loop).

Question 9 [2 Marks]

There have been five plenary lectures so far in this course, and below there is one question from each. You need only answer **two of these five questions.** Each answer should be one or two sentences. If you answer more than two questions, you must indicate which two you wish to be graded; if you do not indicate which, then the first two in order will be graded.

Plenary Lecture 1 — Creative Application for Mobile Devices (Prof. Jonathan Rose)
This lecture discussed the many capabilities of a smartphone, and different apps that were created for research and regular use. One application that was discussed and demonstrated was called 'MyAnkle' and it used an accelerometer, in a wearable device to measure something. What did it measure and why was that useful to measure?

Answer:

Plenary Lecture 2 — Software at Google Scale (Danyao Wang)
Question: Name three things, according to Danyao Wang at Google, that need to be present in the process of software engineering.

Answer:

Plenary Lecture 3 — How Facebook Software is Made (Prof. Michael Stumm)
Describe the nature of the tasks that Professor Stumm's new kind of 'complete' agile software engineer must undertake.

Answer:

Plenary Lecture 4 — An EE's Journey to Software (Prof. Vaughn Betz)
What did the company that installed the MRI machine in a European Hospital ask Professor Betz's company to do to help them with their problem?

Answer:

Plenary Lecture 5 — User Experience Design (Jason Hyde, Plastic Mobile)
Mr. Hyde suggested that one particular user interface 'element' (or method) had become one of the most popular ways of providing navigation around mobile applications. What was that the name of that method, and how did it work?

Answer:

Question 10 [6 Marks]

This question has two, connected, parts. In the first part you will write the code for a function, and in the second part you will write the `main` function that makes use of that function.

(a) Write a function, called `whichBiggerAndHowMuch` that receives two integer values as its input parameters (among other possible input parameters that you're free to choose) and produces two outputs: first, it should return (using the `return` statement) a `bool`-type variable which will be `true` if the first number is greater than or equal to the second number, and `false` otherwise. Second, it should also send back the absolute value of the difference between the larger and the smaller number (i.e. this number must be positive). You can assume that both of the input integers are positive.

Please write your solution here:

(b) Write the `main` function of a C program that asks the user for two integer inputs, and then calls the above function (`whichBiggerAndHowMuch`) to determine which of the two is larger, and the difference. Here are two different executions of the main program that show you what the input and output should look like:

Example Run 1:

```
Enter first and second number: 56 22
First number is bigger by 34
```

Example Run 2:

```
Enter first and second number: 12 16
Second number is bigger by 4
```

Please write your solution here:

Question 11 [6 Marks]

Write a complete C program that prints out the following pattern, as a function of an input value `n`. The following is an example run for `n = 6`:

```
Enter n: 5
@@@@@@
@@    @
@ @   @
@   @ @
@    @@
@@@@@@
```

The following is an example run for `n = 8`:

```
Enter n: 8
@@@@@@@@
@@      @
@ @     @
@   @   @
@    @  @
@     @ @
@      @@
@@@@@@@@
```

Your program must request the number `n` from the user at the beginning, and *you can assume that the value of* `n` *is greater than 2.*

Your program may make use of the following function, which you can assume is written before your code:

```
void printNChar(int numC, char c) {
  for (int i = 1; i <= numC; i++)
    printf("%c", c);
}
```

Please write your solution here and continue on next page:

Please continue your solution to Question 11 on this page:

Question 12 [6 Marks]

$\sin(x)$ can be calculated using the Maclaurin series:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \cdot$$

where $n!$ is the factorial of n, defined as $n! = n \times (n-1) \times (n-2) \times \ldots \times 3 \times 2 \times 1$.

Write a complete C program that first prompts the user for a double value x, and then computes $\sin(x)$ using the first 10 terms in the Maclaurin series, and finally prints it with 6 digits after the decimal point. Here is an example run of your program:

```
Enter a value: 3.1415926
result = 0.000000
```

Please write your solution here and continue on next page:

Please continue your solution to Question 12 on this page:

Question 13 [7 Marks]

Write a complete C program that reads an English sentence from user input, and prints out "`Found ECE!`" if the sentence contains three consecutive characters that spell `ECE`. You can assume that the sentence ends with a `'.'` character. Your program should read in one character after another until it reads an 'E' followed immediately by a 'C' followed immediately by an 'E', at which time it should print out "`Found ECE!`". The program should not print anything if `ECE` is not found in the sentence. You are not allowed to use an array in your solution.

Here is an example run of your program:

```
Please enter a sentence: This course is taught by ECE professors.
Found ECE!
```

Please write your solution here and continue on next page:

Please continue your solution to Question 13 on this page:

Question 14 [7 Marks]

Write a complete C program that prompts the user repeatedly for a sequence of up to 10 integer values. After receiving all 10 values, **or** if the user enters `0`, the program will stop prompting for more values. You can assume that the user enters at least one value before entering `0`. Your program will complete the following three tasks, in the order as given below, using the values the user entered:

- Print the total number of values entered;
- Print all the values in the order that the user entered them;
- Print whether the values are entered in ascending order, i.e., the next value is either greater than or equal to the previous one. For example, {`3, 4, 7, 7`} is a sequence of values in ascending order, but {`3, 4, 7, 6`} is not.

Hint: you will need to use an array for this question.

Here are a few example runs of your program.

Example run 1:

```
Enter a value (0 to stop): 1
Enter a value (0 to stop): 2
Enter a value (0 to stop): 3
Enter a value (0 to stop): 4
Enter a value (0 to stop): 7
Enter a value (0 to stop): 7
Enter a value (0 to stop): 8
Enter a value (0 to stop): 9
Enter a value (0 to stop): 10
Enter a value (0 to stop): 11
There are a total of 10 numbers.
The values you entered are: 1 2 3 4 7 7 8 9 10 11
The values are in ascending order.
```

Example run 2:

```
Enter a value (0 to stop): 3
Enter a value (0 to stop): 5
Enter a value (0 to stop): 5
Enter a value (0 to stop): 7
Enter a value (0 to stop): 0
There are a total of 4 numbers.
The values you entered are: 3 5 5 7
The values are in ascending order.
```

Example run 3:

```
Enter a value (0 to stop): 2
Enter a value (0 to stop): 1
Enter a value (0 to stop): 3
Enter a value (0 to stop): 0
There are a total of 3 numbers.
The values you entered are: 2 1 3
The values are not in ascending order.
```

Example run 4:

```
Enter a value (0 to stop): 1
Enter a value (0 to stop): 0
There are a total of 1 numbers.
The values you entered are: 1
The values are in ascending order.
```

Please write your solution here and continue on next page:

Please continue your solution to Question 14 on this page:

This page has been left blank intentionally. You may use it for answers to any question in this examination.