UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

APS 105 — Computer Fundamentals
Midterm Examination
October 20, 2011
6:15 p.m. – 8:00 p.m.
(105 minutes)

Examiners: J. Anderson, T. Fairgrieve, B. Li, G. Steffan, A. Veneris

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. You may use any math function that you have learned, as necessary.

The examination has 13 pages, including this one.

Circle your lecture section (**one mark deduction** if you do not correctly indicate your section):

L0101	**L0102**	**L0103**	**L0104**	**L0105**	**L0106**
Li	Fairgrieve	Anderson	Steffan	Fairgrieve	Veneris
Monday 2 PM	Monday 9 AM	Monday 11 AM	Monday 11 AM	Monday 4 PM	Monday 4 PM

Full Name: _____

Student Number: _____ UTORid: _____

MARKS

1	2	3	4	5	6	7	8	9	10	11	12	13	14	Total
/3	/3	/3	/4	/4	/3	/8	/8	/4	/12	/12	/12	/12	/12	/100

Question 1 [3 Marks]

Write a single C statement that declares a `bool` variable named `myBool` and sets its value to `true` if and only if the value stored in an integer variable named `myInt` is an even positive number. You may assume that the variable `myInt` has been declared and initialized already to some value.

Question 2 [3 Marks]

The *golden ratio conjugate* Φ is defined as $\Phi = \frac{\sqrt{5}-1}{2}$. This mathematical constant is used in different fields of computer engineering such as numerical analysis, combinatorial algorithms, etc. Write a single C statement that declares a constant variable named `PHI` and sets `PHI` to be equal to the value of Φ given above. You may make use of functions defined in `math.h`.

Question 3 [3 Marks]

Suppose we have the following declaration and initialization of variables:

```
double a = 23.0, b = 20.0, c = 15.0, d = 8.0, e = 40.0;
```

What is the value for the following expression?

```
(int) (a + b / (c - 5.0)) / ((d + 7.0) / (e - 37.0) / 2.0)
```

Question 4 [4 Marks]

Let `i` be an `int` variable with value 5, `j` be an `int` variable with value 0, and `k` be a `bool` variable set to `false`. State whether the following expressions evaluate to `true` or `false`.

Expression	Answer
`( (i >= 3) && (j == (int) 0.0) )`	
`( ((i != 3) || (j != 4)) == !((i == 3) && (j == 4)) )`	
`!( (i-4) <= 0 )`	
`( k && (sin(42.0) == 0.1372) )`	

Question 5 [4 Marks]

Write a single C statement that declares an `int` type variable named `lotto`, and initializes it to be an odd random integer between 601 and 649 (inclusive).

Question 6 [3 Marks]

What will be printed by the following C program?

```
int main (void)
{
    int i;
    for(i = 0; i < 5; i++)
        printf("%c", 'a'+i);

    return 0;
}
```

Question 7 [8 Marks]

Given the following C program:

```
#include <stdio.h>

int main (void)
{
  int a = 1;
  int b = 2;

  while (a <= 45)
  {
    printf("%d\n", a);
    a += b;
    b++;
  }

  return 0;
}
```

What is the output from an execution of this C program?

Question 8 [8 Marks]

Given the following C program:

```
#include <stdio.h>

int main (void)
{
  int row, col;
  for (row = 1; row <= 5; row++)
  {
    for (col = 1; col <= 9; col++)
    {
      if ((col > row) && (col <= (9-row)))
        printf(" ");
      else
        printf("*");
    }
    printf("\n");
  }

  return 0;
}
```

What is the output from an execution of this C program?

Question 9 [4 Marks]

For each of the following parts, determine the number of times that the word `Chicken` will be printed. Place your answer in the appropriate box.

(a)

```
for (int i=0; i < 10; i++)
    for (int j=0; j < 10; j++)
        if (i < j)
            printf("Chicken\n");
```

Will print `Chicken` [ ] times.

(b)

```
for (int i=0; i >= 0; i++)
    for ( ; i < 100; )
        while (i != 0)
            printf("Chicken\n");
```

Will print `Chicken` [ ] times.

Question 10 [12 Marks]

Write a program that prompts the user to enter two dates and then indicates which date comes earlier in the calendar. Assume that the user enters valid dates.

Here is a sample output from an execution of the program:

```
Enter first date (day/month): 20/10<enter>
Enter second date (day/month): 22/9<enter>
22/9 is earlier than 20/10
```

Here is another sample output from an execution of the program:

```
Enter first date (day/month): 20/10<enter>
Enter second date (day/month): 20/10<enter>
The two dates are the same.
```

```
#include <stdio.h>
int main (void)
{









}
```

Question 11 [12 Marks]

Write a program that finds the largest positive number in a list of numbers entered by the user. The program must prompt the user to enter numbers one by one. When the user enters 0 or a negative number, the program must stop prompting the user for numbers and display the largest positive number that was entered. Note that the length of the list of positive numbers could be arbitrarily large. You may assume that at least one positive number will be input.

Here is a sample output from an execution of the program:

```
Enter a number: 60<enter>
Enter a number: 4.89<enter>
Enter a number: 100.62<enter>
Enter a number: 75.2295<enter>
Enter a number: 0<enter>
The largest positive number entered was:  100.620000
```

Notice that the numbers aren't necessarily integers.

```
#include <stdio.h>
int main (void)
{









}
```

Question 12 [12 Marks]

Write a function named `pythagoreanTriples` that accepts a positive integer x as its parameter, and prints one triple (x, y, z) such that:

(a) $x^2 + y^2 = z^2$;

(b) $y > 0$ and $y <= 100$;

(c) $y < z$;

(d) x, y, z are all positive integers.

If no triple exists that satisfies all four conditions, then the function prints `"No solution exists."`. Assume that the argument value for x is a positive integer.

For example, if we compile and run the `pythagoreanTriples` function with the following `main()` program:

```
int main (void)
{
  pythagoreanTriples(3);
  pythagoreanTriples(12);
  pythagoreanTriples(20);
  pythagoreanTriples(49);
  return 0;
}
```

the following will be printed:

```
(3, 4, 5)
(12, 5, 13)
(20, 15, 25)
No solution exists.
```

Place your solution to this question on the next page.

Place your solution to Question 12 on this page.

```
void pythagoreanTriples (int x)
{









}
```

Question 13 [12 Marks]

In the indicated position in the program below, define a function called `rotate` that rotates the values pointed to by the three pointer-to-integer parameters in such a way that after executing the function, the first has the original value of the third, the second has the original value of the first, and the third has the original value of the second.

Example 1:

```
Please input three integers: 3 4 5
After rotation, the list of integers is: 5 3 4
```

Example 2:

```
Please input three integers: 9 3 7
After rotation, the list of integers is: 7 9 3
```

Program:

```
#include <stdio.h>

// define rotate HERE:
void rotate (int *a, int *b, int *c)
{









}

int main (void)
{
 int x,y,z;

 printf("Please input three integers: ");
 scanf("%d %d %d",&x,&y,&z);

 rotate(&x,&y,&z);

 printf("After rotation, the list of integers is: %d %d %d\n",x,y,z);

 return 0;
}
```

Question 14 [12 Marks]

Amicable numbers are two different positive integers such that the sum of the proper exact divisors of each is equal to the other number. A proper exact divisor is a positive integer exact divisor of a number, other than the number itself. For example, the proper exact divisors of 6 are 1, 2, and 3.

220 and 284 are amicable numbers. The reason is as follows: the proper exact divisors of 220 are 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, and 110, the sum of which is 284. The proper exact divisors of 284 are 1, 2, 4, 71, and 142, the sum of which is 220.

Write a complete C function called `areAmicable`, the prototype of which is given below, that accepts two numbers of type `int` as parameters, named `m` and `n`. The function must return the Boolean type `bool`. The function should return `true` if `m` and `n` are amicable numbers, and should return `false` otherwise. You may assume that the arguments passed to the function are indeed positive integers.

```
bool areAmicable (int m, int n)
{









}
```

This page has been left blank intentionally. You may use it for answers to any questions.