

**UNIVERSITY OF TORONTO  
FACULTY OF APPLIED SCIENCE AND ENGINEERING**

**APS 105 — Computer Fundamentals  
Midterm Examination  
October 28, 2008  
12:20 p.m. – 1:50 p.m.**

**Examiners: Jason Anderson, Tom Fairgrieve, Baochun Li**

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 or the back of each page to complete your answer. Please indicate clearly on which question(s) you are answering.

You must use the C programming language to answer programming questions.

The examination has 11 pages, including this one.

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

**L0101**      or      **L0102**      or      **L0103**      or      **L0104**  
Fairgrieve (M14)      Anderson (M9)      Fairgrieve (M11)      Li (M11)

Full Name: \_\_\_\_\_

Student Number: \_\_\_\_\_ ECF Login: \_\_\_\_\_

**MARKS**

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

**Question 1 [3 Marks]**

Given the following C code:

```
int j, sum = 0;
for (j = 0; j < 10; j++)
    sum += j;
```

Rewrite the C statements so that they compute the same quantity but use a while loop instead of a for loop.

```
int j = 0, sum = 0;
while (j < 10)
{
    sum += j;
    j ++;
}
```

**Question 2 [3 Marks]**

Write a single C statement that rounds the value of a double variable `averageGrade` to four digits after the decimal point, and stores the result in `averageGrade`. For example, if `averageGrade` is 3.1415927, after the statement is executed, `averageGrade` is 3.1416. You may use any of the mathematical functions discussed in the course.

Solution:

```
averageGrade = rint(averageGrade * 10000) / 10000;
```

**Question 3 [3 Marks]**

Write a single C statement that uses `malloc` to create an array named `list` that holds 100 values of type `double`.

Solution:

```
double * list = (double *) malloc(sizeof(double) * 100);
```

**Question 4 [3 Marks]**

Show the output produced by the following C program fragment.

```
int i = 1, j = 2, k = 3;
printf("%d", (i + 5) % (j + 2) / k);
```

Solution: 0

**Question 5 [4 Marks]**

Write a single C statement that determines an even random integer number between  $-50$  and  $50$  inclusive, and then assigns it to an `int` variable `r` that has already been declared.

Solution:

```
r = rand() % 51 * 2 - 50;
```

**Question 6** [4 Marks]

Suppose that `a` is a one-dimensional `int` array and `p` is a pointer to `int` variable. Assuming that the assignment `p = a` has just been performed, which of the following expressions are illegal because of mismatched types? Of the remaining expressions, which are true?

Solution:

Expression	Answer
<code>p == a[0]</code>	illegal
<code>p == &amp;a[0]</code>	true
<code>*p == a[0]</code>	true
<code>p[0] == a[0]</code>	true

**Question 7 [8 Marks]**

Given the following C program:

```
#include <stdio.h>

int main (void)
{
    int r, c, n = 3, v[] = {8, 3, 9, 6, 5};
    int toeplitz[n][n];

    for (r = 0; r < n; r++)
    {
        for (c = 0; c < n; c++)
        {
            toeplitz[r][c] = v[(n-1) - (r-c)];
            printf("%d ", toeplitz[r][c]);
        }
        printf("\n");
    }
    return 0;
}
```

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

Solution:

```
9 6 5
3 9 6
8 3 9
```

**Question 8 [8 Marks]**

Given the following C program:

```
#include <stdio.h>

int main (void)
{
    int i = 0, j = 21;
    for (; i < j; i += 2, j -= 3)
        printf("%d\n", i * j);
    printf("%d,%d\n", i, j);
    return 0;
}
```

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

Solution:

0  
36  
60  
72  
72  
10,6

**Question 9 [8 Marks]**

If you have a certain number of US dollars and wish to convert them to Canadian dollars, you could use the Canadian dollar to US dollar exchange rate (for example: 1 Canadian dollar = 0.81 US dollar). Write a complete C program that prompts its user for the current Canadian dollar to US dollar exchange rate (e.g. 0.81) and a value in US dollars, and then prints the value in Canadian dollars.

Solution:

```
#include <stdio.h>
int main (void)
{
    double exchangeRate, cdn, usd;

    printf("Enter the exchange rate (1 CAD = ? USD): ");
    scanf("%lf", &exchangeRate);
    printf("Enter the value in US dollars: ");
    scanf("%lf", &usd);
    cdn = usd / exchangeRate;
    printf("The value in Canadian dollars is %f.\n", cdn);

    return 0;
}
```

**Question 10** [14 Marks]

Write a complete C function named `findItem`, the prototype of which is given below, that returns the pointer to the first occurrence of an integer value `item` (specified as a parameter) in an array `list`. If the value `item` is not found in the array, the function returns the null pointer. Both the array and its size are specified as parameters.

Solution:

```
int * findItem(int * list, int size, int item)
{
    int i;
    for (i = 0; i < size; i ++)
        if (*(list + i) == item)
            return list + i;
    return NULL;
}
```

**Question 11** [14 Marks]

Write a complete C function named `printTriangle` that prints the pattern shown below, with the given number of rows passed in as a parameter. For example, if the number of rows is 7, the function should produce the following output.

```

    *
   ***
  *****
 *****
*****
*****
*****
*****
```

If the parameter value is not positive, the function should print nothing.

Solution:

```
void printTriangle(int rows)
{
    int i, j;
    for (i = 1; i <= rows; i++)
    {
        for (j = 1; j <= rows - i; j++)
            printf(" ");
        for (j = 1; j <= i * 2 - 1; j++)
            printf("*");
        printf("\n");
    }
}
```

**Question 12** [14 Marks]

Write a complete C function named `swapLargest` that accepts two parameters: (1) an array of integers and (2) the array's length. The function finds the largest element in the array, and swaps the largest element with the element in the last array position.

Solution:

```
void swapLargest(int list[], int length)
{
    int largest = 0;
    int i, temp;

    for (i = 1; i < length; i++)
        if (list[i] > list[largest])
            largest = i;

    temp = list[largest];
    list[largest] = list[length-1];
    list[length-1] = temp;
}
```

**Question 13** [14 Marks]

The value of the mathematical constant  $e$  can be expressed using the infinite series:

$$e = 1 + 1/1! + 1/2! + 1/3! + \dots$$

Write a complete C program that approximates  $e$  by approximating the value of

$$1 + 1/1! + 1/2! + 1/3! + \dots$$

Rather than adding an infinite number of terms, your program should continue adding terms until the value of a term is less than 0.001. Your program should print the approximation to  $e$  and the number of terms used to determine the approximation. The *terms* in the series are 1, 1/1!, 1/2!, and so on.

Solution:

```
int main(void)
{
    const double TOLERANCE = 0.001;
    double sum = 0.0, term = 1.0;
    int n = 0;

    while (term >= TOLERANCE)
    {
        sum = sum + term;    // accumulate the term
        n = n + 1;          // determine next term
        term = term / n;
    }

    printf("The value of e is approximately %f.\n", sum);
    printf("The number of terms in the sum is %d.\n", n);

    return 0;
}
```

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