

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

**APS 105 — Computer Fundamentals  
Final Examination  
April 21, 2015  
9:30 a.m. – 12:00 p.m.  
(150 minutes)**

**Examiners: B. Korst, K. Phang**

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

The **Table of Common C functions and the ASCII Character Table** can be found on the last page of this exam.

**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 required** to write `#include` directives in your solutions, whenever needed. You may use any math function that you have learned, as necessary.

The examination has 14 pages, including this one.

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

Student Number: \_\_\_\_\_ UTORID: \_\_\_\_\_

**MARKS**

1	2	3	4	5	6	7	8	9	10	11	Total
/10	/2.5	/7.5	/7	/4	/4	/5	/12	/6	/14	/8	/80

**Question 1 [10 Marks]**

Indicate whether each of the following statements is *True* or *False*. Write your answer in the table given.

- (a) **True or False:** A function can be defined inside another function.
- (b) **True or False:** This statement is correct: `double x[3] = {3.2, 6.1, 7.5, 9.4}`
- (c) **True or False:** Whenever a `for` loop is executed, the statement in the body of hte loop is always executed at least once.
- (d) **True or False:** These two statements are equivalent: `(j++/2 - j)` and `(++j/2 - j)`
- (e) **True or False:** When arrays are used as parameters in function calls, they can be passed either by value or reference.
- (f) **True or False:** The value of the expression `(int) (6.5 + (double)11 / 3);` is 10.0
- (g) **True or False:** The size of a statically allocated array cannot be increased.
- (h) **True or False:** Size of `short integer` and `long integer` would vary from one platform to another.
- (i) **True or False:** If `list` has been declared as an array and `p` is a pointer to `list`, then `&list[4]` is a synonym for `list+4` and `p+4`.
- (j) **True or False:** The statement `m %= n/5;` is equivalent to `m = m% (n/5);`

**ANSWER**

a	b	c	d	e	f	g	h	i	j

**Question 2 [2.5 Marks]**

What is the output of the following code:

```
#include<stdio.h>
int main()
{
    char c = 'A';
    int i, a[]={18, 12, 8, 11, 4};
    for (i=0; i<5; i++)
        printf("%c", (char)(c + a[i]));
    return 0;
}
```

**Answer:**

**Question 3 [7.5 Marks]**

Find the compile-time errors in the following program. Correct the errors, rewriting or adding any lines as needed.

```
/* This function adds two integers */
#include<stdio.h>;
int main()
{
    int integer1, integer2, sum; /* declarations */

    printf("Enter first integer\n");
    scanf("%d", &integer1); //reads first integer

    printf("Enter second integer\n");
    scanf("%d", &integer2); //reads second integer

    sum = int1 + int2; /* adding */
    printf("Sum is %d\n", &sum); //printing sum

    return 0; /* termination
}
```

**Question 4 [7 Marks]**

Write a function called quadx for calculating the roots of a quadratic equation,  $x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$  and  $x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$ . The function must generate two results,  $x_1$  and  $x_2$ . It returns  $x_1$  and places the value of  $x_2$  into a given address in memory. Variables  $x_1$ ,  $x_2$  are double, and  $a$ ,  $b$ , and  $c$  are int, all previously declared. You may use math functions. The main function calls quadx in the following manner:

```
x1 = quadx(a, b, c, &x2);
```

**Answer:****Question 5 [4 Marks]**

Write a function named testDiv that has two parameters of type int, and returns a type bool. The function accomplishes a test to see if the larger of the two inputs is exactly divisible by the other.

**Answer:**

**Question 6 [4 Marks]**

The following code compiles and runs without any errors. After the run is complete, what are the values of the variables on the table? Write them on the table.

```
#include <stdio.h>

/* helper function */
int func(int *k, int j)
{
    int i, m = 1;

    for (i = 0; i < j; ++i) {
        m += *k;
        *k -= 1;
    }

    return m;
}

int main ()
{
    int d, n = 5, h = 3;

    d = func(&n, h) ;

    return 0;
}
```

Variable	Value
i	
*k	
d	
n	

**Question 7 [5 Marks]**

Write below *what is printed* by the following C program after execution. Make sure you show the steps you took in your tracing to reach the answer.

```
int theNumber( int a[], int z)
{
    if (z == 0)
        return a[0];
    else
        return (a[z] + theNumber (a, z-2));
}

int main (void)
{
    int i, m[11];

    for (i=0; i<=10; i++)
        m[i] = i+2;

    printf("The number is %d \n", theNumber(m, 6));
    return 0;
}
```

**Answer:**

**Question 8 [12 Marks]**

Complex numbers are particularly useful in Engineering. The multiplication of two complex numbers  $m_1 = a + jb$  and  $m_2 = c + jd$  results in a third complex number  $m_3 = x + jy$ , whose real ( $x$ ) and imaginary ( $y$ ) components are calculated as follows:

$$x = ac - bd \quad y = bc + ad$$

Write a full C program to implement this complex number multiplication. The only library you are allowed to use is `stdio.h`. Your program is *required* to have:

- The real and imaginary coefficients for both numbers (all `int`) entered by the user;
- The real and imaginary coefficients for all complex numbers held in a data type called `Complex`, which you must define;
- A function called `multiply`, whose prototype is `Complex multiply (Complex, Complex);`
- Your function called from within the main function;
- The result of the multiplication (i.e., complex number  $m_3$ ) printed to the screen with both its real and imaginary components.

A sample output would be:

```
Enter real and imaginary values for first number: 4 3
Enter real and imaginary values for second number: 2 1
The result is: 5 + j 10
```

**Answer:**

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

**Question 9 [6 Marks]**

A certain matrix multiplication function multiplies matrix A and B, and stores the result in matrix Result. Your task is NOT to write the multiplication function, but to declare the matrices, allocate and deallocate memory for them as described below.

Each of the three matrices is located in *sequential* blocks of memory (a sequence of contiguous memory addresses per matrix). The data held in each matrix is of type float. The dimensions are to be passed to the function as int parameters. Matrix A has dimensions (m by n), and Matrix B has dimensions (p by q). Access to the elements of each matrix will be done via pointers (not as two dimensional array indexes).

Write the declarations for matrices A, B and Result (1 line). Remember, they are *not* to be declared as two dimensional arrays.

Write the memory allocation for matrices A, B and Result (3 lines). Do not use for loops.

Write the memory de-allocation for matrices A, B and Result (3 lines). Do not use for loops.

**Question 10 [14 Marks]**

The data type presented below defines a node of a linked list which will hold information pertaining to athletes for the upcoming PanAm games. You will need the information below for all remaining questions. However, the answer to one question does *not* affect the answers to the others.

```
typedef struct athlete{  
    char name[20];  
    char nation[20];  
    char sport[20];  
    struct athlete* next;  
}ATHLETE;
```

The head and the tail of the linked list have been declared as

```
ATHLETE *ListHead, *ListTail;
```

Write a function `insertList` to insert a new node. Make sure that this function:

- allocates memory as needed;
- inserts the new node at the end of the list, unless it is the first node (check it!).

**Answer:**

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

**Question 11 [8 Marks]**

Assuming the database using the linked list on the previous question is filled with all information, write a function called `checkNation` to count how many athletes from a certain nation will be present in the games. You may use `strcmp` in your function. The call to the function should be:

```
x = checkNation(ListHead, "Canada");
```

Note: you do not need the answer to the previous question to solve this question.

**Answer:**

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

REFERENCE: Some Common C functions

<b>Math functions – #include &lt;math.h&gt;</b>	<b>Value returned</b>
<code>sqrt(x)</code>	$\sqrt{x}$
<code>pow(x,y)</code>	$x^y$
<code>log(x)</code>	$\ln x$
<code>exp(x)</code>	$e^x$
<code>fabs(x)</code>	$ x $
<code>fmax(x,y)</code>	maximum of arguments
<code>fmin(x,y)</code>	minimum of arguments
<code>floor(x)</code>	largest integer $\leq x$ (as a double)
<code>ceil(x)</code>	smallest integer $\geq x$ (as a double)
<code>sin(x)</code>	$\sin x$ (in radians)
<code>cos(x)</code>	$\cos x$
<code>atan(x)</code>	$\arctan x$ ( $x$ in radians)
<b>String functions – #include &lt;string.h&gt;</b>	
<code>int strlen( char *s )</code>	Returns length of string.
<code>int strcmp( char *s1, char *s2 )</code>	Compare.
<code>char *strcpy( char *s1, char *s2 )</code>	Copy s2 to s1. Returns pointer to s1.
<code>char *strcat( char *s1, char *s2 )</code>	Concatenate s2 to s1. Returns pointer to s1.
<code>char *strstr( char *s1, char *s2)</code>	Find leftmost occurrence of s2 in s1.
<b>Standard utility functions – #include &lt;stdlib.h&gt;</b>	
<code>double atof( char *nvalstr )</code>	Convert numeric string to double.
<code>int atoi( char *nvalstr )</code>	Convert numeric string to int.
<code>long atol( char *nvalstr )</code>	Convert numeric string to long.
<code>int rand()</code>	Generates pseudorandom integer.

ASCII Character Table (Excerpt)

Char	Decimal	Char	Decimal
0	48	A	65
1	49	B	66
2	50	C	67
3	51	...	...
4	52	X	88
5	53	Y	89
6	54	Z	90
7	55	...	...
8	56	a	97
9	57	b	98
...	...	c	99
		...	...
		x	120
		y	121
		z	122