Walton Karl Avillanosa
CMSC 21 Removal Exam

Part 1: Multiple Choice (52 points - 2 points per question)

1. Which statement about C is true?
   (A) Lines beginning with a # are processed at execution time.
   (B) Good comments improve execution-time performance.
   (C) Reading a value into a memory location destroy the previous value.
   (D) **none of the above**
2. Which of the following is true?
   (A) The equality operators associate right to left.
   (B) @ is an address operator.
   (C) _list1 is an invalid identifier.
   (D) **none of the above**
3. Which is a multiple selection structure?
   (A) **switch** (B) if · · · else
   (C) for (D) while
4. Which is an example of a ternary operator?
   (A) <= **(B) ?:**
   (C) ++ (D) none of above
5. How many times will the following program print Happy!?
   i = 1;
   while ((i *= 2))<2000) printf("Happy!");
   (A) 8 (B) 9 **(C) 10** (D) none of above
6. Which is the printf conversion specification for long int?
   (A) %ld (B) %lh (C) %lu (D) none of the above
7. Which is equivalent to if (n != 8)?
   (A) if !(n = 8) (B) if !(n - 8) **(C) if (n > 8 || n < 8)**
   (B) (D) none of above
8. Which is illegal type in C?
   (A) int short unsigned (B) long double
   (C) long unsigned long **(D) none of above**
9. int add(int); is an example of a function
   (A) data type (B) mode type
   (C) prototype (D) procedure type
10. Consider int b[3][2] = {{1}, {2, 3}, {4}}, what is the value of b[1][1]?
    (A) 2 (B) 3
    (C) 4 **(D) none of above**
11. Given a[3] = {1, 2, 3} and b[3] = {1, 2, 4}. What is the value of a[a[0]] + b[a[1]]?
    (A) **4** (B) 5
    (C) 6 (D) 7
12. Which function is used to seed a new random number sequence?
    (A) **rand** (B) seed
    (C) setrand (D) srand
13. A recursive function is a function that
    (A) returns itself (B) takes a function as an argument
    (C) calls itself (D) is inside of another function
14. Unless otherwise specified, an individual array element is passed _____ and an entire array is passed _____.
    (A) call-by-value, call-by-value (B) call-by-reference, call-by-reference
    (C) call-by-reference, call-by-value **(D) call-by-value, call-by-reference**
15. Which is a correct to pass int a[10][10] into a function f?
    (A) f(a) **(B) f(a[][])**
    (C) f(a[][10]) (D) f(a[10][10])
16. Which type of variables is not destroyed on exit from the function?
    (A) automatic (B) extern
    (C) register **(D) static**
17. Which value does strcmp("Aloha!", "hello!") possibly return?
    (A) **-1** (B) 0
    (C) 1 (D) none of above
18. Which function can be used to get a string from stream?
    (A) fgetc (B) fget
    (C) fgets (D) none of above
19. If a = 2.0 and b = 2.0, what is printed by printf("%.2f", pow(sqrt(a + b), 3))?
    (A) **8.00** (B) 27.00 (C) 64.00 (D) none of the above
20. Which is true?
    (A) **An array can contain data items of different data types.**
    (B) An array size can be changed after declaration.
    (C) The subscript for the last element of an array is the array size.
    (D) none of the above
21. The strcmp function will return a ____ value if its arguments are equal.
    (A) negative **(B) 0** (C) 1 (D) none of the above

22. Which of the following is not a repetition structure?

(A) for (B) do · · · while

(C) while **(D) switch**

23. Which is a benefit of functions?

(A) Reduce programming errors. (B) Divide and conquer.

(C) Make a program more efficient. (D) none of the above

24. Which is true?

(A) **All variables defined inside a function are local variables.**

(B) Parameters are required for any function.

(C) A function must have a return type.

(D) none of the above

25. Assume hello is a character array. Which of the following operations does not produce a string?

(A) hello[] = {'h', 'e', 'l', 'l', 'o'}; (B) hello[] = {'h', 'e', 'l', 'l', 'o', '\0'};

(C) hello[] = " "; (D) hello[] = "hello";

26. An array b is pointed by *p. With which pointer expression b[3] can be referenced?

(A) p + 3 (B) b[p + 3]

(C) *b[p + 3] (D) *(p + 3)

Part II. Question and Answer

1. (16 points) Identify and correct the errors in each of the following statements:

(a) (3 points) char *str = {"happy"}; str[1] = "e"; str[2] = "l";

(b) (3 points) mul (double x, y) { double x, y; return x * y; }

> Errors: return

> Correction: mul { double x,y; double x*y; };

(c)

```
int *xp;  //references array x
void *vp = NULL; int num;
int x[5] = {1, 2, 3, 4, 5}; vp
= arr;
```

a. ++xp;

b. num = xp; //use pointer to access first element (assume xp is initialized)

c. num = *xp[1]; //assign element 1 (value 2) to num

d. ++x;

2. (10 points) Write a code for the following:

3. The square root of a positive number can be approximated by the following iterative method

$$y_{n+1} = \frac{1}{2}\left(y_n + \frac{x}{y_n}\right)$$

Where x is the number entered by the user and y_{n+1} is the next guess for the square root of x, computed using its old value y_n and x. Since an initial guess is required, we set $y_0 = 1$.

See the example below:

x	y	x/y	$\frac{1}{2}(y_n + \frac{x}{y_n})$
3	1	3	2
3	2	1.5	1.75
3	1.75	1.73429	1.73214
3	1.73214	1.73196	1.73205
3	1.73205	1.73205	1.73205

Use a loop to iterate until the absolute value of $y_{n+1} - y$ is less than or equal to the tolerance, given by the variable *tol = 0.00001*.

Use the `fabs` function to find the absolute value of a `double`, from the `<math.h>` header.

Prompt the user to enter x and display the final approximation.

```c
#include <stdio.h>
#include <math.h>

int main(void)
{
    float yn, i;
    double x, result;

    printf("Enter the number you wish to find the square root of
    printf("\n");
    scanf("%f", &yn);

    x = yn/2;

    for (i = 0; i < 100; i++)
        x = x - (((x*x) - yn)/(2*x));
     result = fabs(x);
    printf("The square root of %.0f is %.4f.\n", yn, x);

    return 0;

}
```

4. (15pts) Write the following functions using the following structure for a point.

```
struct Point{
        double x, y;
}p;
```

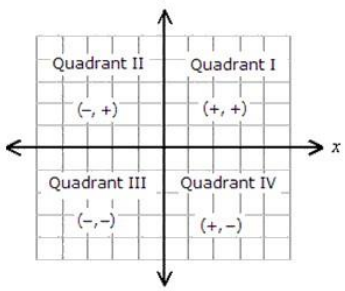
```c
#include <stdio.h>
#include <math.h>
        struct point { double x; double y; };
        int main(void) {
            struct point test;
            test.x = .25; test.y = .75;
            printf("(%6.4f, %6.4f)\n", test.x, test.y);
            return 0;
        }
```

b. (4 points) A function that passes two points and returns the distance of them

c. (4 points) A function that passes two points and returns the slope.

Slope:

$$m = \frac{y_1 - y_2}{x_1 - x_2} = \frac{y_2 - y_1}{x_2 - x_1}$$



Distance between two points:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$