CISC 105 Practice Final Exam

Name ________________________ Login name ________________

Section: ______ TA ________

General Instructions

- DO **NOT** PUT YOUR **SSN** ON ANYTHING!
- DO NOT WRITE YOUR NAME ON ANY PAGE EXCEPT THIS ONE!
- Turn off any noise making device, especially **CELL PHONES**. You may lose up to one letter grade if your device disturbs the peace of the exam.
- You have 50 minutes. **Pace yourself,** and pay attention to the point values.
- The exam is 61 points multiple choice, and 89 points programming and short answer.
- Do not add features that are not required by the problem. For example, if the instructions don't say anything about user input, then your program should not take user input. If you aren't sure, ask.
- Do problems you are confident about first. If you finish the problems you know, write what you do know about other problems to gain partial credit; but erroneous information may detract from that credit or irritate the grader, so don't make stuff up.
- Read *all* the directions *carefully* on each problem.
- Often writing a fast, rough version of a program in English or pseudocode will make your C coding faster and more accurate. It also enables me to give partial credit in some circumstances.
- You may assume that input will not produce errors for the code described, unless the questions say otherwise.
- Do not do unnecessary testing. For example, testing for both `x < 0` and `x >= 0` instead of using one test and then `else` would be considered unnecessary testing.
- Have fun!

Error definition

Errors do not always create problems in output, but should be considered errors anyway, as we do in class. This means errors include, but are not limited to:

- incorrect format specifiers
- incorrect number of parameters
- reading/writing invalid memory
- reading uninitialized ("garbage") memory
- other compilation errors
- other runtime errors

Errors do not include type coercion in arithmetic (for example, assigning a double to an int) unless that impairs the correctness of a program.

For multiple choice questions, if the code completes without error, select the answer that shows what is printed. If the code has an error (see the front page for definition of an error) select error. If no other answer is correct, select "none of the above".

1. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
  int a[4] = {13, 14, 15, 19};
  printf("%d",a[2]);
  return 0;
}
```

(a) `0` (b) `15` (c) `14` (d) `13` (e) error

2. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
  int b[3] = {13, 15, 17, 19};
  printf("%d",b[3]);
  return 0;
}
```

(a) `17` (b) `19` (c) `0` (d) `'\0'` (e) error

3. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
  int a[5] = {16, 17, 18, 19, 20};
  printf("%d",a[5]);
  return 0;
}
```

(a) `19` (b) `20` (c) `0` (d) `none of these` (e) error

4. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
  int data[5] = {3, 1, 5, 9};
  printf("%d",data[4]);
  return 0;
}
```

(a) `-1` (b) `0` (c) `1` (d) `9` (e) error

For the next four questions, assume the variables will have the following addresses:

name	address
x	FB00
y	FB04
p	FB08

5. (2 pts) If the variable `x` is of type `int`, which of the following expressions would print the address of that variable?

   (a) `printf("%a", &x);`

   (b) `printf("%d", x);`

   (c) `printf("%p", FB00);`

   (d) `printf("%p", &x);`

   (e) none of the above

6. (2 pts) What is the output?

```
int x = 5;
int *p;
*p = &x;
*p = 5;
printf("%d\n", x);
```

   (a) 2

   (b) 5

   (c) 0

   (d) error

   (e) none of the above

7. (2 pts) What is the output?

```
int x = 2;
int *p = &x;
printf("%d\n", *p);
```

   (a) 2

   (b) x

   (c) 0xFB00

   (d) error

   (e) none of the above

8. (2 pts) What is the output?

```
int x = 5;
int *p = &x;
*p = *p - *p;
printf("%d\n", x);
```

   (a) NULL

   (b) 0

   (c) 5

   (d) error

   (e) none of the above

9. (2 pts) What is the output?

```
int x = 2;
int *p = &x;
int y = *p;
printf("%d\n", y);
```

(a) 2
(b) x
(c) 0xFB00
(d) error
(e) none of the above

For the next **four** questions, assume x and y are declared as **integers**. Choose the letter that shows the value that would be in the variable x. See the definition of "error" at the front of the exam. The fragments are unrelated (each question is separate from the others).

10. (2 pts)

```
y = 7.5;
x = 2 * y;
```

(a) `0` (b) `15` (c) `14` (d) `error` (e) none of the above

11. (2 pts)

```
double z = 10;
x = z / 4;
```

(a) `1.5` (b) `2` (c) `2.5` (d) `error` (e) none of the above

12. (2 pts)

```
x = 5 % 4;
```

(a) `1` (b) `0` (c) `1.2` (d) `4` (e) none of the above

13. (2 pts)

```
x = 0 || -1;
```

(a) `0` (b) `1` (c) `true` (d) `error` (e) none of the above

14. (2 pts) What is the last index of `w` that is initialized after the declaration shown?

```
char w[6] = "dirt";
```

(a) 2
(b) 3
(c) 4
(d) none of the above
(e) error

15. (2 pts) Which of the following is a correct way to change **c** after the code shown?

```
    int main(){
        char c;
        ...
```

(a) `char c == 's';`
(b) `c = "s";`
(c) `c = 'r' + 1;`
(d) `strcpy(c, "s");`
(e) none of the above

16. (2 pts) What is the output?

```
#include <stdio.h>
int main(void)
{
  char a[10][40] = {"albatross","spam","nugent","that"};
  printf("%c",a[3][1]);
  return 0;
}
```

(a) h
(b) u
(c) t
(d) none of the above
(e) error

17. (2 pts) Which of the following would correctly allocate space for the string "spam" at the first space in **words**?

```
    int main(){
        char * words[5];
        ...
```

(a) enough space is already allocated
(b) `words[1] = malloc(strlen("spam"));`
(c) `words[0] = allocate(strlen("spam"));`
(d) `words[0] = malloc(strlen("spam") + 1);`
(e) none of the above

18. (2 pts) Which of the following assigns a random value to an integer x?

    (a) `srand(x)`
    (b) `x = time(NULL);`
    (c) `x=rand();`
    (d) `rand(x);`
    (e) none of these

19. (3 pts) Assume x contains a random number. Which of the following puts x in the range 0-100, inclusive?

    (a) `x = x % 100;`
    (b) `x %= 101;`
    (c) `x.range(100);`
    (d) `x.range(101);`
    (e) none of these

20. (3 pts) How many numbers will binary search examine to find one member of a sorted list of 55,000,000 numbers?

    (a) approximately 26
    (b) approximately 28
    (c) approximately 36
    (d) approximately 38
    (e) none of the above

21. (2 pts) Which of the following is true?

    (a) malloc returns a pointer to space on the heap
    (b) dereferenced parameters are placed on the heap
    (c) structs cannot be placed on the heap
    (d) pointer variables are not placed on the stack
    (e) none of the above

22. (2 pts) What is wrong with this code fragment?

```
1)   int main(){
2)       char temp[10];
3)       char * w = temp;
4)       strcpy(w, "spam");
```

    (a) w is the wrong type for strcpy
    (b) w does not point to allocated memory
    (c) line 2 should have asterisk removed
    (d) w can only hold an address, not an array name
    (e) nothing is wrong

23. (2 pts) Selection sort works in a series of passes over an array. Choose the answer that shows how this array will appear after the first TWO passes of selection sort (as discussed in class). Place larger elements to the right.

    original array: 5 1 9 4 8 3 6 2

    (a) 1 5 4 8 3 6 2 9
    (b) 5 1 9 4 8 3 6 9
    (c) 5 1 4 8 3 6 2 9
    (d) 1 5 4 9 3 8 2 6
    (e) none of the above

24. (2 pts) Selection sort works in a series of passes over an array. Choose the answer that shows how this array will appear after the first TWO passes of selection sort (as discussed in class). Place larger elements to the right.

    original array: 6 4 1 8 3 2 5 9

    (a) 6 4 1 5 3 2 8 9
    (b) 4 6 1 8 2 3 5 9
    (c) 4 1 6 3 2 5 8 9
    (d) 1 2 3 4 5 6 8 9
    (e) none of the above

25. (1 pts) True or false: Selection sort and binary search, as coded in class, are both recursive functions.

    (a) true
    (b) false
    (c) none of the above

26. (1 pts) True or false: The following function is recursive:

```
int f(){
   recursive("f");
   return 0;
}
```

    (a) true
    (b) false

27. (1 pts) True or false: The following function is recursive:

```
int f(int i){
   if (i < 7)
       return f(i * 2);
   else
       return 0;
}
```

    (a) true
    (b) false

28. (1 pts) True or false: The following function is recursive:

```
int f(int i){
   if (i <= 7)
       return 2 * f(i - 1);
   else
       return 0;
}
```

(a) true
(b) false

29. (1 pts) True or false: The following function is recursive:

```
int f(int n){
   int product = 1;
   for(n; n > 1; n--)
      product *= n;
   return product;
}
```

(a) true
(b) false

30. (2 pts) When a struct is passed as a parameter to a function using a function call, C uses

(a) pass by association
(b) pass by value
(c) pass by assignment
(d) pass by reference
(e) none of the above

```
typedef struct {
    double highTemp;
    double lowTemp
    char date[10];
} Record;
```

31. (2 pts) Assume two such Record structs, **day1** and **day2**, have been declared, and **day2** has been initialized. Which of the following will take the values from day2 and put them into day1?

(a) day2=day1;
(b) day1=day2;
(c) strcpy(day2, day1);
(d) strcpy(day1, day2);
(e) it must be done with a while loop
(f) none of the above

32. (2 pts) Which of the following functions could take one of the structs shown above as a parameter?

(a) Record print(double highTemp, double lowTemp, char date[10]);
(b) void print(struct Record k);
(c) int print(Record k);
(d) void print(Record k[]);
(e) none of the above

Use this definition for the following four questions.

```
typedef struct {
  int height;
  double circumference;
  char name[30];
} Tank;
```

33. (3 pts) Write one line of code to declare a tank. Do not initialize any values in the tank.

34. (3 pts) Use the tank you declared in problem 33 and change the height to 14.

35. (3 pts) Use the tank you declared in problem 33 and change the name to "George".

36. (3 pts) Declare an array of 30 tanks.

37. Assume all necessary libraries are included. Use these declarations to answer the following six questions.

```
char *pc;
char word1[10];
char word2[10];
char data[] = "the show jumped the shark";
```

(6 pts) Show code that puts the first word from **data** into **word1**. Your code must use **strtok**. You may use up to 2 lines of code.

(4 pts) Continue your code from above and use **strtok** again to make **pc** point to the second word from **data**.

Assume all necessary libraries are included. Use these declarations to answer the following six questions.

```
int * ip;
int i = 1;
int iPtr;
```

38. (3 pts) Before you write the answer to this question, read the following question and review the declarations above. Then write code that allocates space on the heap for a single integer.

39. (3 pts) Show how to place the integer 3 in the space from problem 38.

40. (4 pts) Write the prototype for a function that takes a string parameter and returns a struct of the kind you declared in problem 33.

41. (4 pts) Write the prototype for a function to print an entire array of structs of the kind you declared in problem 33.

42. (4 pts) Using the integer named **i** declared in main, show how to pass **i** to the function from problem 41 by value.

43. (25 pts) Consider the following main() and its output:

```
int main(){

    int i;
    int a[] = {4,5,3,2,1};
    swapMaxWithEnd(a, 5);
    printArray(a, 5);

    return 0;
}
```

output:

```
% a.out
1 4 3 2 5
```

Write the single function swapMaxWithEnd() so it can be used in a sorting algorithm like selection sort. Write the function so that the code above produces the output shown, but your code should work for any integer array. Note that this function is not the same as any function we wrote for selection sort in class. First it finds the highest number in the array, then it swaps it with the last number in the array.

44. (20 pts) There are **four** blank areas in the program below. Fill them in so that the program behaves as shown. Please think before you code.

```
% a.out
Enter a word: eggogog
g count is: 4
```

```
#include <stdio.h>

/*
 * Program to count the number of g's in a word
 * Terry Harvey CISC 105 section 98 TA Janet
 */

int main(){
    int i = 0;
    int count = 0;
    char word[20];

    /* Prompt user for input */



    /* Get a single word from user input */



    /* Count the number of times the letter g occurs in the string */











    /* Print the result. */



    return 0;
}
```