All multiple choice questions are equally weighted. You can generally assume that code shown in the questions is syntactically correct, unless something in the question or one of the answers suggests otherwise. The questions are not intended to require knowledge of advanced or obscure C language features not mentioned in CSE142.
CSE142 Final exam 6/6/01, version A

1. What is output by the following program?

```
#include <stdio.h>
#define N 10

int main( )
{
   int i, j;
   int A[N];

   for(i=0; i<N; i=i+1) {
      A[i] = 2;
      for(j=0; j<i; j=j+1) {
         A[i] = A[i] + 1;
      }
   }
   printf("%d\n", A[3]);
   return 0;
}
```

A. 2
B. 3
C. 5
D. 9
E. 11

2. Which of the following is guaranteed to be true about EOF in C?

A. It is a special data value written at the end of each file.

B. It is a status indicating that the end of a file has been reached.

C. It is defined in the stdio.h header file.

A. A only
B. B only
C. C only
D. A and B
E. B and C

3. Suppose @ and # are operators (in some imaginary programming language that has precedence and associativity rules like C), and suppose you know that the expression
@ @ a # @ b # c

is evaluated as if it had parentheses as follows:

(@ (@ a)) # ((@ b) # c)

Study this information to see if you can tell whether the operators are unary or binary, and something about their associativity.
What must be true?

A. @ is binary, left associative
B. @ is unary, left associative
C. # is unary
D. # is binary, left associative
E. # is binary, right associative

4. What's the output of the following code fragment:

```
#include <stdio.h>

void foo(int d)
{
   int a;

   a = d;
   d = 2 * a;
   printf("%d ", d);
}

void main(void)
{
   int a, d;

   d = 3;
   a = 2;
   foo(3);
   printf("%d %d", a, d);
}
```

A. 6 3 6
B. 6 3 3
C. 6 2 6
D. 3 2 6
E. 6 2 3

5. A palindrome is a word which is spelled the same forwards as it is backwards. Eamples:
ABBA
nun

Below is a function is_palindrome() which determines if a string is a palindrome. This function relies on the recursive helper function recursive_palindrome().

The code for recursive_palindrome is missing a line at the indicated position. Choose the line which correctly completes the function.

```
#define TRUE  1
#define FALSE 0

int is_palindrome(char word[])
{
  int len = strlen(word);

  return recursive_palindrome(word, 0, len - 1);
}


int recursive_palindrome(char word[], int start, int end)
{
  if (start >= end)
    return TRUE;
  else {
    /* MISSING RETURN STATEMENT: CHOOSE ONE THAT WORKS */
  }
}
```

A.
```
return ((word[start] == word[end]) &&
            recursive_palindrome(word, start + 1, end - 1));
```
B.
```
return ((word[start] == word[end]) &&
            recursive_palindrome(word, start, end));
```
C.
```
return (word[start] == word[end]);
```
D.
```
return ((word[start] == word[end]) ||
            recursive_palindrome(word, start + 1, end - 1));
```
E.
```
return (recursive_palindrome(word, start + 1, end - 1));
```

6. What is the value of k at Point A?

```
int i, j, k;

k = 0;

for (i = 0; i < 3; i++) {
   for (j = (3-i); j > 0; j--) {
      k = k + j;
   }
}
/* Point A */
...
```

A. 9
B. 10
C. 12
D. 14
E. 16

7. Here are three consecutive lines from a C program:

```
...
double income;
printf("Enter new income: ");
scanf("%lf", &income);              /* line A */
...
```

Line A is which of the following?
I. An assignment statement
II. Initialization of income
III Declaration of income

A. I only
B. II only
C. III only
D. All of I, II, and III
E. None of I, II, or III

8.

```
typedef struct {
   int x, y;
} Point;

typedef struct {
   int color;
   Point pos;
} ColoredPoint;

ColoredPoint cp;
```

What is the correct code to move cp to the origin of the coordinate system?

A.
```
cp->pos.x = 0;
cp->pos.y = 0;
```
B.
```
cp.pos->x = 0;
cp.pos->y = 0;
```
C.
```
cp->pos->x = 0;
cp->pos->y = 0;
```
D.
```
cp.pos.x = 0;
cp.pos.y = 0;
```
E.
```
cp->pos = {0, 0};
```

9. We want code to set the diagonal entries of a 5 by 5 two dimensional array ("matrix") to 1; all other entries in the matrix are to be left unchanged. Which choice is correct?

Background: The "diagonal" entries of a matrix are those where the column number and row number are the same. The term is only used there are the same number of rows as columns. For example, in the following 3 by 3 matrix:

1 2 3
4 5 6
7 8 9

the diagonal entries have indices [0][0], [1][1], and [2][2], and the diagonal values are 1, 5, and 9.

A.
```
int arr[5][5], i;
for (i = 0; i <= 5; i++)
    arr[i][i] = 1;
```
B.
```
int arr[5][5], i;
for (i = 0; i < 5; i++)
    arr[i][i] = 1;
```
C.
```
int arr[5][5], i;
for (i = 0; i < 5; i++)
    arr[i, i] = 1;
```
D.
```
int arr[5][5], i;
for (i = 0; i <= 5; i++)
    arr[i] = 1;
```
E.
```
int arr[5][5], i, j;
for (i = 0; i < 5; i++)
    for (j = 0; j < 5; j++)
        arr[i][j] = 1;
```

10. Here's a truth table that is not completely filled in. Which of the choices gives the correct values for the last column?

A	**B**	**!(A && !B)**
T	**T**	
T	**F**	
F	**T**	
F	**F**	

A. T
T
T
T

B. F
T
F
F

C. F
T
F
T

D. T
T
F
T

E. T
F
T
T

11. What combination of values of a, b, and c make the following condition evaluate to 1 (or "true")? All variables are integers, although you can think of them as Booleans.

```
(!a || b) && (!b && c)
```

A. a = 0, b = 0, c = 0
B. a = 0, b = 0, c = 1
C. a = 1, b = 1, c = 0
D. a = 0, b = 1, c = 1
E. a = 1, b = 0, c = 1

12. Study the following function. Try to summarize what is does, and then choose the description which matches best.

```
int followingFunction(int A[], int x, int y){
  int i;
  int z = 0;

  for (i=x; i<=y; i++) {
    z = z + A[i];
  }

  return z;
}
```

A. counts the number of elements between the xth and yth element (inclusive).
B. counts the number of elements between the xth and yth element (exclusive).
C. sums the elements from the xth to the yth (exclusive)
D. sums the elements from the xth to the yth (inclusive)
E. multiplies x times y

13. Here are fragments illustrating three proposed solutions to the problem of swapping two integers. Which are valid?

```
/************Proposal A */
void swap (int *a, int *b) {
    int temp;
    temp = *a;
    a = b;
    *b = temp;
}

int main (void) {
    int i, j;
...
    swap (&i, &j);
...

/***********Proposal B */
  void swap (int a, int b) {
    int temp;
    temp = a;
    a = b;
    b = temp;
}

int main (void) {
    int i, j;
...
    swap (i, j);
...

/***********Proposal C */
void swap (int *a, int *b) {
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}

int main (void) {
    int i, j;
...
    swap (&i, &j);
...
```

A. A only
B. B only
C. C only
D. A and B
E. None of A, B, or C

14. Recall that a pre-condition is "a condition assumed to be true before a function call" and a postcondition is "a condition assumed to be true after a function executes." (Hanly & Koffman p.131).

Assuming the following function is correct, what would be a proper precondition for it?

```
double recL (double a, double w) {
    double q;
    q = a / w;
    return q;
}
```

A. a is greater than or equal to 0
B. w is greater than 0
C. q has been properly initialized
D. a and w have different values
E. the return value of the function is different from a

15. Your prof sez that in the following fragment of a function, line A is OK but B will not compile. He's right. What's true that explains this?

```
char courseName[7] = "CSE142";   /*line A*/
...
courseName = "142";    /*line B*/
```

A. It's purely a matter of style.
B. The length of the string in line B is wrong
C. Line A is a declaration
D. Line B is a declaration
E. The contents of an array cannot be changed once it has been assigned a value

16. Recall that \n symbolizes the linefeed character, and \0 symbolizes the null character. Both of these are single characters in memory.

StringA contains:
Hello\n\0

String B contains:
world\n\0

The two are concatenated with the string library call
strcat (StringA, StringB);

What does the resulting string contain?

A. Hello\n\world\n
B. Helloworld\n
C. Hello\0world\0
D. Hello\nworld\n\0
E. Hello\n\0world\n\0

17. Suppose an array contains the following eight values:

-50 30 99 6 12 15 1 14

What is true?

A. The array satisfies the precondition for Linear Search
B. The array satisfies the precondition for Binary Search
C. The array satisfies the postcondition for Selection Sort

A. A only
B. B only
C. C only
D. B and C only
E. All of A, B, or C

18. Let's say we are applying binary search to the array of 6 values

10 18 19 22 30 60

The target is 30. To start the search, L is -1 and R is 6. After the first probe, what happens?

A. L increases
B. L decreases
C. R increases
D. R decreases
E. L and R do not change throughout the algorithm

19. When binary search is applied to an array of size 16, about 4 probes (tests of array elements) are needed to determine if a target value is in the array. Suppose an array has 20000 elements. About how many probes are needed?

A. 15
B. 50
C. 200
D. 500
E. 5000

20. Consider the following declarations and function prototypes:

```
typedef struct {
        char type[50];
        double height;
        int age;
} tree;

int seed( int count, char *kind_of, tree grove[6] );

tree forest[ 27 ];
int height, age;
char type[] = "maple";
tree elm = { "elm", 25.6, 42 };
```

Given the above information, which of the following pairs correctly calls function seed?

A. seed(13, forest[6].type, *forest[4]);
B. seed(elm.age, type, &forest);
C. seed(elm.age, type, forest[0]);
D. seed(forest[13].age, type, forest);
E. seed(age, type[0], &forest[2]);

21. What output would you get if you made the function call
`recursiveFunction(6)?`

```
int recursiveFunction(int num)
{
        if (num == 0)
        {
                return num;
        }
        else
        {
                num = -1 + recursiveFunction(num);
                printf("%d", num);
                return num;
        }
}
```

A. 6543210
B. 654321
C. 666666
D. 0123456
E. There will be an infinite recursion

22. [worth 12 MC questions]
Write a program which reads in information about students from a file, and then prints out certain parts of the information, in order by the students' names.

Before you start, note that there are three parts to the solution, which should be solved in sequence. The final program is part 3, and will use results from the earlier parts.

Read the background information first, then begin solving the parts of the problem. Use good programming style in your solution.

There is a header file called student.h. You have this on a handout. Reading through it now would be a good idea. student.h contains struct and #define definitions as well as function prototypes. Your solution should make use of these when appropriate. You can and should call the functions whose prototypes are given there. Other than compareNames and sortStudents, you may assume that all function prototypes in student.h have been fully and correctly implemented.

1. Implement the function compareNames. Use the prototype for it in the student.h #include file. You may call the string library function strcmp, but no other functions (information about strcmp is in the student.h header file). Write your answer here:

2. Implement the function sortStudents. Do this by modifying the code given below; this is the code for a Selection Sort of integers, exactly as given in the lecture slides (two functions, sel_sort and min_loc). Do NOT start over or rewrite the functions. Simply cross out what needs to be changed and write in your changes. Don't make any more changes than are absolutely necessary. Assume that the swap function works correctly already.

```
void sel_sort (int b[ ],  int n) {

   int k, m;

   for (k = 0; k < n - 1; k = k + 1) {

        m = min_loc(b,k,n);

        swap(&b[k], &b[m]);

   }
}

int min_loc (int b[ ], int k, int n) {

   int j, pos;  /* b[pos] is smallest element */
                        /* found so far */
   pos = k;

   for ( j = k + 1; j < n; j = j + 1)

      if (b[j] < b[pos])

        pos = j;

   return pos;
}
```

3. Write a complete program which reads in student information from a file, and then prints a report of students in alphabetical order, one student per line. For each student, print their name (last name, comma, first name), followed by that student's average quiz score and average homework score. The data about students is contained in a file with the external name of "StudentDataSet.txt". You can call any of the functions in student.h, and you can call printf, but no other functions. Write your answer here:

```
#include "student.h"


int main (void) {









}
```