All multiple choice questions are equally weighted. You can generally assume that code shown in the questions is intended to be syntactically correct, unless something in the question or one of the answers suggests otherwise.

1. What is printed by the following program fragment?

```c
int p_test (int a, int * b) {
    int d;
    d = a - *b;
    a = d;
    *b = d;
    return d;
}

int main (void) {
    int a=1, b=2, c=3, d=4;
    c = p_test (a, &b);
    printf "%d %d %d %d", a, b, c, d);
    ...
}
```

A. 1 2 3 4  
B. -1 -1 -1 -1  
C. 1 -1 -1 4 Correct Answer  
D. -1 -1 -1 4  
E. 1 -1 -1 -1

2. What does the following code print? (Assume the program compiles without error). Trick question! Note the semicolon at the end of the second line.

```c
int i;
for(i = 1; i <= 10; i = i + 1);
printf("%d", i);
```

A. the numbers 1 through 10  
B. the number 1 (only)  
C. the number 10 (only)  
D. the number 11 (only) Correct Answer  
E. can't tell, because i is not initialized

3. What is the output of the following program fragment?

```c
x = 6;
y = 5;
z = 1;
if (x <= y || y <= z)
    printf("Yellow");
else
    printf("Green");
```

A. No output, because the printf doesn't have a % (placeholder).  
B. Can't tell, because  
   (x <= y || y <= z)  
   is not legal C (is a syntax error).  
C. Yellow  
D. Green Correct Answer  
E. YellowGreen
4. What is the output of the following program?

```c
#include <stdio.h>

void F(int a, int b)
{
    int c;
    c = a;
    a = b;
    b = c;
}

int main (void)
{
    int a,b,c;
    a = 5;
    b = 4;
    c = 10;
    F (a, b);
    printf ("a=%d b=%d c=%d", a, b, c);
    return 0;
}
```

A. a=1 b=5 c=10
B. a=5 b=4 c=10 Correct Answer
C. a=5 b=4 c=4
D. a=4 b=5 c=5
E. a=21 b=22 c=23

5. Which of these three program fragments have the same output (apart from spacing)? All variables are ints.

i. printf("1\n2\n3\n4\n5\n6\n7\n8\n9\n10");

ii. for (i = 9; i > 0; i--)
{
    j = i - 10;
    printf("%d\n", -j);
}

iii. i = 1;
    while (i < 10)
    {
        printf("\%d\n", i);
        i++;
    }

A. i. and ii.
B. ii. and iii. Correct Answer
C. i., ii., and iii.
D. None have the same output
E. i. and iii.
6. Consider the following program fragment:

```c
...  
double taxes[3];  
double total;  
```

Choose the best statement:

A. Line A is wrong because of an out-of-bounds error  Correct Answer
B. Line A is correct, but for the sake of efficiency, line A should be rewritten as a for-loop.
C. Line A can be written more efficiently as total += taxes[ ];
D. Line A has a logic error because total is not initialized.
E. There is no logic or syntax error in line A.

7. The following program fragment has its lines numbered for reference along the left-hand side. The code prints a figure that looks like this:

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

Suppose we wanted the output look like this instead:

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

What changes to the code would accomplish this?

A. 3) int row = 4;  Correct Answer
   4) while (row >= 1){
      11) row = row - 1;
   B. 1) #define SIZE (-4)
   C. 5) col = 4;
      6) while (col >= row){
   D. 3) int row = 4;
      11) row = row - 1;
   E. 5) col = 4;
8. What values of a, b, and c (respectively) make the following condition evaluate to "true" (or 1)?

\( !(b \&\& a) \&\& (c \&\& b) \)

I) 0,0,0  
II) 1,1,1  
III) 1,1,0  
IV) 1,0,0

A. I, II, and III only  
B. II and IV only  
C. IV only  
D. There are some combinations of values which make it true, but none are listed among the choices  
Correct Answer  
E. There exists no combination of values make this expression true

9. What is true about compound statements in C?

A. A compound statement must contain at least one statement.  
B. A compound statement cannot contain another compound statement  
C. A compound statement cannot contain any function calls  

A. A only  
B. B only  
C. C only  
D. None of A, B, or C  
Correct Answer

10. If x (an integer variable) has the value 1 at the start of the following code fragment, what is its value at the end?

```c
if (x < 0 || x == 3) {
    x = x - 1;
}
if (x < 2*x && x - 2 < 0) {
    x = x + 1;
}
else {
    x = x + 3;
}
```

A. 0  
B. 1  
C. 2  
D. 3  
Correct Answer  
E. 4
11. Which of the following does not itself cause a change in the sequential control flow of a program in execution?
   A. assignment statement  Correct Answer
   B. function call
   C. return statement
   D. switch statement
   E. if statement

12. A "sentinel" in programming, as we have used it, is...
   A. a statement which checks for an error
   B. a special value on input  Correct Answer
   C. the vertical bar character used in the "or" operation
   D. a peer or colleague who reads a program to check for errors
   E. any warning or error when a program is compiled

13. Among the choices offered, which condition is equivalent to the following (assume the variables are chars):

\[(\text{answer} != 'y') \&\& (\text{option} == 'q')\]

   A. !((\text{answer} != 'n') \&\& (\text{option} != 'q'))
   B. !(\text{answer} == 'n') \&\& !(\text{option} == 'q')
   C. \((\text{answer} == 'y') || (\text{option} != 'q')\)
   D. \((\text{answer} != 'y') || (\text{option} != 'q'))\)
   E. !((\text{answer} == 'y') \&\& (\text{option} != 'q'))  Correct Answer

14. Suppose that the function Snipe begins

\[
\text{void Snipe(int *f, double d, char *ch)\
 int i, j;\
 double x, y;\
 char a, b;\
 ...}
\]

What are the types of the three expressions:

\[(\text{int}) *\text{ch} + i\]
\[*d / *f\]
\[*f < i\]

   A. int*, double, &int
   B. int, illegal, int  Correct Answer
   C. char, double, illegal
   D. illegal, illegal, int
   E. int, illegal, int*
15. The function AddOne can be used to increment a variable. The code for AddOne is:

```c
void AddOne(int *x)
{
    *x = *x + 1;
}
```

Which of the following is a correct implementation of the AddTwo function, which adds two to a variable:

A. ```c
void AddTwo(int *y)
{
    AddOne(&y);
    AddOne(&y);
}
```  
B. ```c
void AddTwo(int *y){  Correct Answer
    AddOne(y);
    AddOne(y);
}
```  
C. ```c
void AddTwo(int *y){
    AddOne(*y);
    AddOne(*y);
}
```  
D. ```c
void AddTwo(int y){
    y = y + 2;
}
```  
E. ```c
void AddTwo(int *y){
    int temp;
    temp = *y;
    AddOne(&temp);
    AddOne(&temp);
}
```  

16. Here is an incomplete truth table for !(P&&Q). If the last column were filled in, how many T’s would that column contain?

<table>
<thead>
<tr>
<th>P</th>
<th>Q</th>
<th>!(P&amp;&amp;Q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

A. 0  
B. 1  
C. 2  
D. 3 Correct Answer  
E. 4
17. [Worth 2 M.C. questions]

Write the function getIntInRange, which is intended to return a value typed in by the user. The function will make sure that the user types in a value that is within a given range of integers.

`getIntInRange` takes two integer parameters (min and max) and returns an integer. The return value should be between min and max, inclusive. For example, if min is 4 and max is 7, then the answer returned must be: 4, 5, 6, or 7. (You may assume that min is never greater than max.)

You should prompt the user for a value, indicating the valid range of inputs. Once the user enters an integer, you must verify that the input is valid. If the input is valid, that value is returned by the function. If not, you should display an error message, and try again (i.e., display a prompt, wait for input, etc.), as many times as is necessary. (You may assume that the user enters only integers; there is no need to try to handle non-integer input).

Sample execution:
Enter a value in the range [4-7]: 3
3 is not in the range [4-7]! Try again.
Enter a value in the range [4-7]: 5

In the sample, min is 4 and max is 7. The function returns 5.

```c
/* Get an integer in a given range from the user */
int getIntInRange(int min, int max){
    int value;
    int done;

    done = FALSE;
    while (done == FALSE){
        printf("Enter a value in the range [%d-%d]:", min, max);
        scanf("%d", &value);
        if (value >= min && value <= max)
            done = TRUE;
        else
            printf("%d is not in the range [%d-%d]! Try again.\n", value, min, max);
    }
    return value;
}
```
18. [Worth 3 M.C. questions. This is a continuation of the preceding problem.]

Write a program that draws facial features. The program repeatedly asks the user for a command (a single character), and performs the action as shown below. You must call getIntInRange in your solution to get the integers from the user.

User enters    Action
q            Program ends
e            Gets an integer in the range [2-6] from the user
             Displays the eyes using the user’s input as the size
n            Gets an integer in the range [0-8] from the user
             Displays the nose using the user’s input as the size
m            Gets an integer in the range [10-16] from the user
             Displays the mouth using the user’s input as the size
all others    Displays an error message “Invalid input”

You can use the following functions to complete the program (i.e., these are written for you; do NOT write these functions, just call them. You should ONLY write main), in addition to the getIntInRange function you have already written:

```c
#include <stdio.h>

int main(void) {
    char cmd;
    int done;
    int size;

    done = FALSE;
    while (! done){
        printf("Enter command: ");
        scanf("%c", &cmd);
        switch (cmd) {
        case 'q':
            done = TRUE;
            break;
        case 'e':
            size = getIntInRange(2, 6);
            drawEyes(size);
            break;
        case 'n':
            size = getIntInRange(0,8);
            drawNose(size);
            break;
        case 'm':
            size = getIntInRange(10,16);
            drawMouth(size);
            break;
        default:
            printf("Invalid Input\n");
        }
    }

    return 0;
}
```

void drawEyes(int size);
void drawNose(int size);
void drawMouth(int size);