1. Output: 12.3 45 128.0
2. Output: 4 11, 6 reasoning: i++ means that the i gets incremented after initialization while ++j means that j gets incremented first before j got initialized, so it would be i=10 and j=6 when the subtraction takes place, 10-6=4
3. Output:6 15 reasoning: i =6 because --i means that i gets decremented first before initialization, when we start solving for j, it would be 3+6*2. To find j, we just follow the order of precedence from left to right in this problem so multiplication gets priority over addition, so 6*2=12, 3+12=15
4. Technically it could have been correct if we are just printing the statement, but because we are also expecting a user input, it is syntactically wrong because it is missing a “scanf” function which would allow you to initialize an input for “n” for this if statement to come out correctly. Thus, in this case, since there is no input, it doesn’t matter what value you put inside the n because it can’t take any input.
5. Let's break the code down
   First condition: i=5>0
   Second condition:5-1 >0 = “j = 4>0”
   If i >0, j>0,
   Third Condition: i=5-1, i=4
   Forth Condition: j = 4-1, j=3
   From the looks of it, j is the loop counter, and the value we are printing out is i.
   First loop, i=5, print 5, i=4, j=3
   Second loop, i=4 Print 4, i=3,j=2
   Third loop, i=3 print 3. i=2, j=1
   Fourth loop, i=2,print 2, i=1 j=0
   Since j is false for this condition j>0, loop ends, so our resulting output is 5 4 3 2.

6. Sum= 0. i=1, If i<=20, add 1 to i. For this problem, i is our loop counter, and it seems that i will continue to count up until i=21, so the loop runs about 20 times.
   Nested if condition: if !(i%3)
   i%3 = true if it’s any nonzero value, and false if it is 0
   !(i%3) = false if its any nonzero and true if it is 0.
   Continue skips the rest of the iteration of the loop, so in this case, skips the sum+=i
   The nested if condition first finds if i modulus 3 is a value lets say true(1), then, the ! will do the opposite of true which will be false(0) . if the nested condition is true, skip that phase and redo the loop counter. If false, we skip the “continue” statement and then we do the sum+=i
   First loop:
   i=1, (1%3) = 1 which is true. !true = false, sum = 0+1 = 1
   i=2, (2%3) =2 which is true. !true = false, sum = 1+2 = 3
   i=3, (3%3) =0 which is false. !false = true, skip
   i=4, (4%3)=1 which is true,, !true = false, sum = 3+4 = 7
   The only times when the continue statement actually executes is when the loop is true, and the only times when it is false, i%3=0, so the only numbers we skip are i=3,6,9,12,15,18,21

So the sum = 1+2+4+5+7+8+10+11+13+14+16+17+19+20 =147
So output for sum = 147

7. Answer: for (n = 0; m > 0; n++, m /= 2); Reasoning: inside the for loop, it allows for multiple expression as long as we separate each condition by a “,”, which is why we can put things like comparing m>0 and n++ incrementations.
8. Corrected code:{

```
int i;
for (i = 0; i < n; i++)
    if (a[i] == 0)
        return true;
return false;
}
```

Reasoning: the criteria ask to return false if ALL elements inside the array are nonzero. The else statement checks the first value that is nonzero and returns false, and so each time a number that is non zero appears, it will return false. We only wanted it to return false after it checks all the elements of the array if it's all nonzero, so we need to write false outside of the for loop because the for loop represents checking each element of the array.

9. Output: i = 5, j = 10, Modified code:

```
#include <stdio.h>

void swap(int *a, int *b);

int main(void)
{
   int i = 5, j = 10;
   swap(&i, &j);
   printf("i = %d, j = %d\n", i, j);
   return 0;
}

void swap(int *a, int *b)
{
   *a = *a + *b;
   *b = *a - *b;
   *a = *a - *b;
}
```

10. #code for calling output:

```
int main() {
   int i;
   scanf("%d", &i);
   pb(i);
   }
   return 0;
```

The function converts the inputted number into binary through recursively calling the function without using a loop.

11,12,13, check files for code, only asked us to provide the c file.