3) (10 pts) DSN (Bitwise Operators)

In this problem we will consider buying a collection of 20 figurines, labeled 0 through 19, inclusive. The figurines come in packages. Each package has some non-empty subset of figurines. We can represent the contents of a single package using an integer in between 1 and $2^{20} - 1$, inclusive, where the bits that are on represent which figurines are in the package. For example, the integer $22 = 2^4 + 2^2 + 2^1$, would represent a package with figurines 1, 2 and 4. Each month, one package comes out. You greedily buy every package until you have all 20 figurines. Write a function that takes in an array of integers, *packages*, and its length, *n*, where *packages[i]* stores an integer representing the contents of the package on sale during month i, and returns the number of months you will have to buy packages to complete the set. It is guaranteed that each figurine belongs to at least one of the packages and that each value in the array packages is in between 1 and 2^{20}-1, inclusive. **<u>For full credit, you must use bitwise operators.</u>**

```
int monthsTillComplete(int packages[], int n) {

    int i = 0, mask = 0;

    while (mask != ((1<<20)-1) ) {
        mask |= packages[i];
        i++;
    }

    return i;

}
```

Grading: 1 pt using an integer to keep track of current items, 4 pts looping until all items are collected (bitshift not necessary, but easier), 4 pts to update current collection (3 pts if no bitwise operator is used), 1 pt increment i, 1 pt return appropriate value.

Max grade for solving correctly with no bitwise operators is 8 out of 10.