

3) (10 pts) DSN (Bitwise Operators)

An organization has 30 groups of employees, labeled as group 0, 1, 2, ..., 29. Each individual employee is assigned to some subset of those groups. The set of groups to which an employee belongs can be stored in a single integer, called the employee's ACCESS CODE, based on the bits of that integer. For example, an employee in groups 0, 3, 13 and 18 would have ACCESS CODE $2^0 + 2^3 + 2^{13} + 2^{18}$ (this is equal to 270345.) There are several shared drives at the organization. Each shared drive is accessible by any employees in a specified set of employee groups. The ACCESS CODE of a drive is specified exactly as that of an employee. If all employees who belong to either groups 2, 3 or 6 should have access to a shared drive, then that drive's access code is $2^2 + 2^3 + 2^6$ (76). An employee with ACCESS CODE 270345 would have access to a drive with ACCESS CODE 76, since the employee is part of group 3, and all employees in group 3 get access to the drive. Write a function that takes in an employee's access code, `empCode`, (as a single integer), an array of integers (`driveCodes`) storing the access codes of every shared drive in the organization, and the length of that array (`numDrives`), and returns the number of the shared drives that the employee with the given access code has access to.

```
int numDrivesAccess(int empCode, int* driveCodes, int numDrives) {  
  
    int res = 0;                                // Grading: 1 pt  
    for (int i=0; i<numDrives; i++)             // Grading: 2 pts  
        if ( (empCode & driveCodes[i]) != 0)    // Grading: 5 pts  
            res++;                               // Grading: 1 pt  
  
    return res;                                // Grading: 1 pt  
  
}
```