1) (10 pts) DSN (Dynamic Memory Management in C)

Suppose we have an array of structures containing information about Cartesian points. The struct shown below contains two integers, one for the x coordinate and one for the y coordinate. For this problem, write a function, createPoints, to create some random Cartesian points with each coordinate set to a random integer in between 0 and 10, inclusive.

createPoints takes in the number of points to be created, *numPoints*. Your function should dynamically allocate an array of *numPoints* CartPoints structs and set each of their x and y coordinates with pseudorandom integer values in between 0 to 10, inclusive. You may assume that the random number generator has been seeded already. Your function should return a pointer to the array that was created and initialized.

```
typedef struct CartPoint {
    int x;
    int y;
} CartPoint;

CartPoint* createPoints(int numPoints) {

    int i;

    // LHS = 1 pt, 3 pts RHS
    CartPoint *somePoints = malloc(sizeof(struct CartPoint) * numPoints);

    for(i=0; i<numPoints; i++) {            // 1 pt
        somePoints[i].x = rand() % 11;      // 2 pts
        somePoints[i].y = rand() % 11;      // 2 pts
    }

    return somePoints;                      // 1 pt

}
```