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

The struct `videogame_t` maintains information about video games that are stored in a store's inventory. As a fan of the 1980s, you would like to get a list of all the games that were produced in between the years 1980 and 1989, inclusive. Write the function below that takes in an array of type `videogame_t`, its length (`n`), and a pointer to a variable that will store the number of games produced in the 1980s. The function should return a newly dynamically allocated array storing a copy of all the information for the games that were produced in the 1980s AND set the variable pointed to by ptrNumGames to the size of the array returned by the function. This copy must be a deep copy, where individual component is copied over, including allocating memory for the copy of the name and copying the name into that new memory. **(Note: Due to the length of code, some of the function has been provided. Don't forget to allocate the appropriate space for each string in each struct!)**

```
//struct representing video game information
typedef struct {
    char * name;
    int year;
    double price;
} videogame_t;

videogame_t* getClassicGames(videogame_t * inventory,
                             int n, int* ptrNumGames) {

    videogame_t* res = malloc(n*sizeof(videogame_t));
    int nG = 0;

    for (int i=0; i<n; i++) {
        if (inventory[i].year >= 1980 && inventory[i].year < 1990) {
            res[nG].name = malloc(sizeof(char)*(1+strlen(inventory[i].name)));
            strcpy(res[nG].name, inventory[i].name);
            res[nG].year = inventory[i].year;
            res[nG].price = inventory[i].price;
            nG++;
        }
    }

    res = realloc(res, nG*sizeof(videogame_t));
    *ptrNumGames = nG;
    return res;
}
```

Grading: 1 pt for loop
3 pts if statement (take off 1 pt if -> is used)
2 pts for malloc of string (take off 1 pt if no room for '\0')
1 pt strcpy (don't give point if they use =)
1 pt copy year
1 pt copy price
1 pt update index (nG)