Memory Worksheet

You are designing a program that manages your book collection. Each book has a title, a list of authors, and a year. Each author you stored their name, birth year, and number of books

Here is the struct information for storing the books and authors,

```
struct author_t
{
    char name[20];
    int birthYear;
    int publishings;
};
```

```
struct book_t
{
    char name[20];
    int year;
    int numAuthors;
    author_t * authors;
};
```

You will read input (from standard input) with the following format:

The first line stores an integer, ***n***, the number of books belonging to your collection. The book information follows. The first line of each book is the book's title (a string 1 to 19 characters no spaces), the year it was published, and ***a***, the number of authors. The following ***a*** lines contains the author description. The author description contains three values the name (a string 1 to 19 characters no spaces), the year of birth, and the total number of books written.

1. Write a segment of code that creates the memory for the list of books (in the form of an array) and authors based on the input format specified above.

2. Write a segment of code that frees the memory that you created.

3. What issues could you run into with updating author information?

4. Which of the following functions have memory violations? Why?

```
typedef struct my_student_t my_student_t;

struct my_student_t
{
    int id;
    char name[20];
};

int * fun1(int n)
{
    int * tmp;
    tmp = (int *) malloc(sizeof(int) * n);
    *tmp = 7;
    return tmp;
}

int fun2()
{
    int * tmp;
    (*tmp) = 7;
    return *tmp;
}

int * fun3()
{
    int * tmp;
    (*tmp) = 7;
    return tmp;
}

my_student_t * fun4()
{
    my_student_t * tmp;
    tmp = (my_student_t *) malloc(sizeof(my_student_t));
    tmp->id = 0;
    tmp->name[0] = ‘\0’;
    return tmp;
}

int fun5() {
    int * tmp = (int *) calloc(1, sizeof(int));
    free(tmp);
    return *tmp;
}
```