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

This problem relies on the following *Player* and the *Team* struct definitions:

```
typedef struct Player
{
  char pname[50];    //player's name
  char country[50]; //player's country
  int age;
} Player;
```

```
typedef struct Team
{
    char tname[50];   // team's name
    Player *players; // all players on the team
    int numPlayers;   // number of players on the team
} Team;
```

We are making a team of players from multiple countries. There is a text file containing the details of a team, where the first line of the file contains the team name, followed by a single space, followed by the number of players on the team, *N*. The next *N* line contains the data for *N* players. Each player line contains three tokens, each separated by a space: the player name, country, and age (as an integer). Each team name, player name, and country will be a single-word string (no spaces) with a maximum length of 49. Here is a sample file:

```
NewKnights 5
Hannan USA 22
Mabel India 21
Samarina Bangladesh 21
Tamsen USA 21
Susan Mexico 22
```

Write a function that takes a file pointer and then returns a pointer to a dynamically allocated Team struct with all the information loaded into it. You can assume that the file is already opened in read mode and ready to read from the beginning of the file. Do not worry about closing the input file with *fclose()* when you finish reading it. Assume the function that opened the file and called *createTeam()* will close the file.

```
Team *createTeam (FILE *fp) {

    Team* res = malloc(sizeof(Team));        // 1 pt

    // 2 pts
    fscanf(fp, "%s%d", res->tname, &(res->numPlayers));

    // 2 pts
    res->players = calloc(res->numPlayers, sizeof(Player));

    // 4 pts - 1 pt loop, 1 pt each read/scanf
    int i;
    for (i=0; i<res->numPlayers; i++)
        fscanf(fp, "%s%s%d", res->players[i].pname,
               res->players[i].country, &(res->players[i].age));

    // 1 pt
    return res;
}
```