

**Problem X.** For each of the following questions, write 1-10 lines of code to accomplish the given task. Do not write functions.

- a. The structure array `sal` is dimension `1xN` and has two fields, `movie` and `rating`. The values of `rating` are guaranteed to be of type `double`. Find the `movie` of the structure within `sal` that has the max `rating` and store it in `maxRating`.

`sal ->`

<code>movie: 'Harry Potter'</code> <code>rating: 10</code>	<code>movie: 'Fudge'</code> <code>rating: 2</code>	<code>movie: 'Aragon'</code> <code>rating: 5</code>
---	---	--

`maxRating -> 'Harry Potter'`

- b. The structure array `sa2` is dimension `1xN` and has 4 different fields. Sort `sa2` by the values in the second fieldname alphabetically. Note: the values in the second fieldname are guaranteed to be type `char`.

`sa2 ->`

sport: 'soccer' viewers:400 player: 'Mesi' wins: 2	sport: 'tennis' viewers: 320 player: 'Nadal' wins: 5	sport: 'football' viewers: 920 player: 'Brady' wins: 30
---	---	--

`sa2 (after) ->`

sport: 'football' viewers: 920 player: 'Brady' wins: 30	sport: 'soccer' viewers:400 player: 'Mesi' wins: 2	sport: 'tennis' viewers: 320 player: 'Nadal' wins: 5
--	---	---

- c. The structure array `studentArray` has four fields: `name`, `age`, `creditHours`, and `fun`. A student's overall score is determined by the sum of their `age`, `creditHours`, and `fun`, which are all guaranteed to be doubles. Create a new field called `studentScore` and put the corresponding overall score of each structure in for the value. Then delete the `age` field from the structure array.

`studentArray` ->

<code>name: 'Joe'</code> <code>age: 12</code> <code>creditHours: 56</code> <code>fun: 2</code>	<code>name: 'Mary'</code> <code>age: 21</code> <code>creditHours: 2</code> <code>fun: 50</code>	<code>name: 'Lexi'</code> <code>age: 40</code> <code>creditHours: 220</code> <code>fun: 0</code>
---	--	---

`studentArray` (after) ->

<code>name: 'Joe'</code> <code>creditHours: 12</code> <code>fun: 56</code> <code>studentScore: 70</code>	<code>name: 'Mary'</code> <code>creditHours: 21</code> <code>fun: 2</code> <code>studentScore: 73</code>	<code>name: 'Lexi'</code> <code>creditHours: 40</code> <code>fun: 220</code> <code>studentScore: 260</code>
---	---	--

## SOLUTION

- a. The structure array `sa1` is dimension `1xN` and has two fields, `movie` and `rating`. The values of `rating` are guaranteed to be of type `double`. Find the `movie` of the structure within `sa1` that has the max `rating` and store it in `maxRating`.

```
ratings = [sa1.rating]
[~, index] = max(ratings)
maxName = sa1(maxMask).movie
```

- b. The structure array `sa2` is dimension `1xN` and has 4 different fields. Sort `sa2` by the values in the second fieldname alphabetically. Note: the values in the second fieldname are guaranteed to be type `char`.

```
allNames = fieldnames(sa2)
[sorted, ind] = sort(allNames)
second = sorted{2}
sortField = {sa2.(second)}
[~, ind] = sort(sortField)
sa2 = sa2(ind)
```

- c. The structure array `studentArray` has four fields: `name`, `age`, `creditHours`, and `fun`. A student's overall score is determined by the sum of their `age`, `creditHours`, and `fun`, which are all guaranteed to be `doubles`. Create a new field called `studentScore` and put the corresponding overall score of each structure in for the value. Then delete the `age` field from the structure array.

```
ages = [studentArray.age]
hours = [studentArray.creditHours]
funz = [studentArray.fun]
total = ages + hours + funz
for x = 1:length(studentArray)
    studentArray(x).studentScore = total(x)
end
studentArray = studentArray.rmfield(studentArray, 'age')
```