Chapter 7 Cell Arrays and Structures

Exercise 7.3  (structure version)

You have been hired by a used-car dealership to modify the price of cars that are up for sale. You will get the information about a car, and then change its price tag depending on a number of factor. Write a function called UsedCar that takes in a structure with the following fields:

- `make` a string of the manufacturer
- `year` a number that correspond to the year of the car
- `cost` the market price of the car
- `miles` the number of miles clocked
- `accidents` the number of accidents the car has been in

Make sure your function should return a structure with all the above fields.

Here are the changes you must make:
1) add 2000 to the cost if the car has clocked less than 20000 miles;
2) subtract 2000 if it has clocked more than 100000 miles;
3) reduce the price by 1000 for every accident.
Chapter 7 Cell Arrays and Structures

Exercise 7.3

You have been hired by a used-car dealership to modify the price of cars that are up for sale. You will get the information about a car, and then change its price tag depending on a number of factor. Write a function called UsedCar that takes in a structure with the following fields:

make,    year,  cost,  miles,  accidents

make     a string of the manufacturer
year     a number that correspond to the year of the car
cost     the market price of the car
miles    the number of miles clocked
accident the number of accidents the car has been in

You function should return a structure with all the above fields.

Here are the changes you must make:
1) add 2000 to the cost if the car has clocked less than 20000 miles;
2) subtract 2000 if it has clocked more than 100000 miles;
3) reduce the price by 1000 for every accident.

clear, clc

archive(1)=makCar(make,     year,  cost, miles, accidents)
archive(1)=makeCar(’Fiat’, 1999, 2000, 250000, 1);
archive(2)=makeCar(’Toyota’, 1997, 7500, 90000, 2);
archive(3)=makeCar(’Ford’, 2012, 15500, 12000, 0);
archive(4)=makeCar(’Citroen’, 2001, 9500, 130000, 1);
archive(5)=makeCar(’Mercedes’, 2007, 27500, 150000, 0);
archive(6)=makeCar(’Audi’, 2003, 19500, 190000, 2);
EmptyCar =makeCar(’ ', 0, 0, 0, 0);

flds = fieldnames(archive);

print_arch(archive);
archive = put_inOrder(archive);
print_arch(archive);

archive = New_Prices(archive)

disp(’new prices are’)
print_arch(archive);

function print_arch(archive)

flds=fieldnames(archive);
for i=1:length(archive)
    Mak = archive(i).(flds{1});
    Yea = archive(i).(flds{2});
    Pri = archive(i).(flds{3});
    Mil = archive(i).(flds{4});
fprintf(’Manufacturer %10s - Year %d - Price %d \n’, Mak, Yea, Pri)
end
end
function archo=put_inOrder(archi)
EmptyCar = makeCar( ' ', 0, 0, 0, 0);
archo = archi;
flds = fieldnames(archo);
goal=true;
while goal
    goal=false;
    for i=1:length(archo)-1
        c1= archo(i). (flds{2});
        c2= archo(i+1). (flds{2});
        if c1>c2
            EmptyCar = archo(i);
            archo(i) = archo(i+1);
            archo(i+1)= EmptyCar;
            goal =true;
        end
    end
end
end

function archive= New_Prices(archive)
flds=fieldnames(archive);
for i=1:length(archive)
    cm=archive(i). (flds{4}); % miles
    Cp=archive(i). (flds{3}); % price
    ac=archive(i). (flds{5}); % accidents
    if cm<20000
        archive(i). (flds{3})=Cp+2000;
    elseif cm>100000
        archive(i). (flds{3})=Cp-2000;
    end
    if ac>0
        archive(i). (flds{3})=archive(i). (flds{3})-(1000.*ac);
    end
end
function Car = makeCar(make, year, cost, miles, accidents)
Car.Make = make;
Car.Year = year;
Car.Cost = cost;
Car.Miles = miles;
Car.Accidents = accidents;
end
Manufacturer       Fiat - Year 1999 - Price 2000
Manufacturer       Toyota - Year 1997 - Price 7500
Manufacturer       Ford - Year 2012 - Price 15500
Manufacturer       Citroen - Year 2001 - Price 9500
Manufacturer       Mercedes - Year 2007 - Price 27500
Manufacturer       Audi - Year 2003 - Price 19500

Manufacturer       Toyota - Year 1997 - Price 7500
Manufacturer       Fiat - Year 1999 - Price 1000
Manufacturer       Citroen - Year 2001 - Price 9500
Manufacturer       Audi - Year 2003 - Price 19500
Manufacturer       Mercedes - Year 2007 - Price 27500
Manufacturer       Ford - Year 2012 - Price 15500

new prices are
Manufacturer       Toyota - Year 1997 - Price 5500
Manufacturer       Fiat - Year 1999 - Price -1000
Manufacturer       Citroen - Year 2001 - Price 6500
Manufacturer       Audi - Year 2003 - Price 19500
Manufacturer       Mercedes - Year 2007 - Price 25500
Manufacturer       Ford - Year 2012 - Price 17500

>>
Exercise 7.3  
(cell array version)

```matlab
% clear, clc
% make, year, cost, Km, accidents)
A = {'Toyota', [1997, 7500, 90000], 2};
B = {'Ford', [2012, 15500, 12000], 0};
C = {'Citroen', [2001, 9500,130000], 1};
D = {'Mercedes', [2007, 27500,150000], 0};
E = {'Audi', [2003, 19500, 19000], 2};
F = {'Empty', [0, 0, 0], 0};

V = {A B C D E F}  % define the cell array of all cars
% sorts the cars, from the cheapest to the expensive
val = true;
while val
    val = false;
    for in = 1:length(V) - 1
        W1 = V{in};  % extract the cell array describing one car
        itemx = W1{1};  % extract the manufacturer
        item1 = W1{2};  % extract the array
        cost1 = item1(2);  % cost
        W2 = V{in+1};  % extract the cell array describing one car
        itemy = W2{1};  % extract the manufacturer
        item2 = W2{2};  % extract the array
        cost2 = item2(2);  % cost
        if cost1>cost2
            V{in} = W2; V{in+1} = W1; val = true;
        end
    end
end
for in = 1:length(V)
    W1 = V{in};  % extract the cell array describing one car
    itemx = W1{1}  % extract the item
    item1 = W1{2};  % extract the item
    cost1 = item1(2);  % cost
end
```

```
V = Columns 1 through 2

{1x3 cell}    {1x3 cell}

Columns 3 through 4

{1x3 cell}    {1x3 cell}

Columns 5 through 6

{1x3 cell}    {1x3 cell}

itemx = Empty    cost1 = 0
itemx = Toyota    cost1 = 7500
itemx = Citroen    cost1 = 9500
itemx = Ford    cost1 = 15500
itemx = Audi    cost1 = 19500
itemx = Mercedes    cost1 = 27500

>>
Chapter 7 Cell Arrays and Structures
Exercise 7.2

It turns out that since you’ve became experts on rating clothing, Acme Clothing Company has hired you to rate their clothes. Clothes are now represented as structures instead of vectors with the fields (all of which are numbers between 0 and 5):

- condition
- color
- price
- matches
- comfort

Acme has a much simpler way of rating their clothes than you used before:

\[
\text{rating} = 5 \times \text{condition} + 3 \times \text{color} + 2 \times \text{price} + \text{matches} + 9 \times \text{comfort}
\]

You have a script called makeClothes.m that will create a structure array called rateClothes that contains clothes structure. You are to write a script called rateClothes that will add a Rating field and a Quality field to each of the structures in the acmsClothes array. The Rating field in each structure should contain the rating of that particular article of clothing. The Quality field is a string that is:
- ‘premium’ if the Rating is over 80,
- ‘good’ if the Rating is over 60,
- ‘poor’ if the Rating is over 20,
- ‘liquidated’ for anything else.

```
clear, clc
% acmeClothes = makeClothes(condition, color, price, matches, comfort)
acmeClothes(1) = makeClothes(5, 5, 35, 5, 3);
acmeClothes(2) = makeClothes(4, 5, 27, 5, 4);
acmeClothes(3) = makeClothes(3, 2, 15, 5, 1);
acmeClothes(4) = makeClothes(0, 3, 7, 5, 4);
acmeClothes(5) = makeClothes(5, 4, 5, 5, 5)
EmptyClothes = makeClothes(0, 0, 0, 0, 0)
flds = fieldnames(acmeClothes)

for i = 1:length(acmeClothes)
    item = acmeClothes(i).(flds{6})
end

EmptyClothes = acmeClothes(1);
acmeClothes(1) = acmeClothes(2);
acmeClothes(2) = EmptyClothes;

for i = 1:length(acmeClothes)
    item = acmeClothes(i).(flds{6})
end
```
function acmeClothes = makeClothes(condition, color, price, matches, comfort)

acmeClothes.Condition = condition;
acmeClothes.Color     = color;
acmeClothes.Price     = price;
acmeClothes.Matches   = matches;
acmeClothes.Comfort   = comfort;

[rating quality]=rateClothes(condition,color,price,matches,comfort);

acmeClothes.Rating    = rating;
acmeClothes.Quality   = quality;
end

function [rating quality]=rateClothes(condition,color,price,matches,comfort)

rating    = 5*condition+3*color+2*price+matches+9*comfort;
if rating >= 80
    quality = 'premium';
elseif rating >=60
    quality = 'good';
elseif rating >=20
    quality = 'poor';
else
    quality = 'liquidated';
end