```
function dish = yelp(filename, category, val)
[num, txt, raw] = xlsread(filename);
headers = raw(1,:);
% find the position of the column stored in the variable category
mask = strcmp(headers,category); (2 pts)
column = raw(:,mask);
 % sort the column of numbers extracted by the mask
[sorted, ind] = sort(cell2mat(column(2:end))); (3 pts)
raw = raw(ind,:);
mask2 = sorted < val;
% delete the rows that are less than val
raw([false mask2],:) = []; (3 pts)
% extract the columns that have prices
prices = cell2mat(raw(2:end,4:6));  (2 pts)
avg = mean(prices,2);
% make a new column where each row is the average of the prices with
% header 'Average Price'
newcol = [‘Average Price’; num2cell(avg)]; (3 pts)
% Append your new column to the array
raw = [raw newcol]; (2 pts)
% extract the signature dish of the top restaurant ranked by
% category
dish = raw{2,3}; (2 pts)
% write to file ending with '_sorted.xlsx'
xlswrite([filename(1:end-5) ‘_sorted.xlsx’], raw);(3 pts)
end
```

Blank 1 (2):
strcmp(headers,category)
or find(strcmp(headers, category));
Give 1 point for strfind()

Blank 2 (3):
sort(cell2mat(column(2:end)))
Give 1 point for indexing
Give 1 point for cell2mat()
Give 1 point for sort()

Blank 3 (3):
raw([false mask2],:) = [];
Give 2 points if forget to account for header

Blank 4 (2):
raw(2:end,4:6)
Give 1 point if indexed entire column

Blank 5 (3):
[‘Average Price’; num2cell(avg)]

1 point for header
1 point for num2cell()
1 point for avg

Blank 6 (2):

```
[raw newcol]
```

No partial

Blank 7 (2):

```
raw{2,3};
```

Give 1 point if () instead of {}

Blank 8 (3):

```
xlswrite([filename(1:end-5) '_sorted.xlsx'], raw);
```

Give 2 points for error in file name