1) What is the output of the program below?

```python
def reorg (mylist):
    unused_items = mylist[:]
    new_list = []
    while unused_items != []:
        item = unused_items[0]
        if item < 10:
            new_list = [item] + new_list
        elif item >= 10:
            new_list = new_list + [item]
        unused_items.remove(item)
    return(new_list)

print(reorg ([9,5,10,12,4,6]))
```
2) a) Write a function called `best_times` that takes a list containing a runner name followed by lists of their times for each event and returns a list containing the runner’s name and a list of their best times for each event. Your function should not modify the original list. For example:

If the original list is:

```python
[‘Saia’, [2.56, 2.30, 2.78, 2.27], [0.23, 0.21, 0.26], [1.39, 1.42]]
```

The result should be:

```python
[‘Saia’, [2.27, 0.21, 1.39]]
```
b) You need to write a function called `update_and_find_best` that will help keep track of runner’s times as new races are being run. The function should return a list of lists. The first list should be an updated list of the runner’s name and all their times for each race type (one time has been added to one of the race time lists, this looks like the input to the function above). The second should be the list of that runner’s best times. When a new race is run the times come back as a list that looks like so:

\[
[\text{race type}, [\text{racename1, racename2, racename3, etc]}, [\text{time1, time2, time3, etc}]]
\]

You can assume that the order of the racer names matches the order in which the times are reported. For example:

\[
[1, ['Saia', 'Sam', 'Hector'], [2.28, 2.57, 2.34]]
\]

2.28 is Saia’s time, 2.57 is Sam’s time, etc..

You should make use of the function you wrote above. Note that there are useful list functions to use, for instance index and append.