Question **1**

Not yet answered

Marked out of 3.00

*Write a **Task** class so that the following code works as specified by the comments. Each comment refers to the line(s) below it. Handle the case where a task is added as a subtask for itself. Specifications or tests are not required **[3p]**.*

```
# Prepare fp, learn py and learn js are tasks
fp = Task("prepare fp")
py = Task("learn py")
js = Task("learn js")
# Here we add two subtasks for py
py.add(Task("types")).add(Task("data model"))
# We also add a subtask for js
js.add(Task("types"))
# 'py' and 'js' are subtasks of fp
fp.add(py).add(js)
# This will recursively search for subtasks of fp named "types"
for t in fp.search("types"):
# Print out the Task
        print(t)
```

Question **2**

Not yet answered

Marked out of 2.00

Analyse the time and extra-space complexity of the following function **[2p]**.

```
def f(l):
    if len(l) == 1:
        return l[0]
    if l[0] == 0:
        return 0
    return l[0] * f(l[1:])
```

Maximum file size: 2MB, maximum number of files: 1

Files

Question **3**

Not yet answered

Marked out of 4.00

Write the specification, Python code and test cases for a ***recursive function*** that uses the ***divide and conquer*** method to calculate the sum of the even numbers found on even positions in a list. The function should return ***None*** if no suitable numbers are found. For the input [**2**, 2, **4**, 5, **6**, 4, 13, 4, **10**], the result is 22. You may divide the implementation into several functions according to best practices. Specify and test all functions **[4p]**.

Maximum file size: 2MB, maximum number of files: 1

Files

You can drag and drop files here to add them.