```python
# -*- coding: utf-8 -*-
# File name: problem1.py
import math
from projectq import MainEngine
from projectq.ops import X, Y, Z, H,  S, T, CX, CZ, Rx, Ry, Rz, Measure, All
from projectq.meta import Loop, Compute, Uncompute, Control
from projectq.cengines import (MainEngine,
                               AutoReplacer,
                               LocalOptimizer,
                               TagRemover,
                               InstructionFilter,
                               DecompositionRuleSet)
import projectq.setups.decompositions
from hiq.projectq.backends import SimulatorMPI
from hiq.projectq.cengines import GreedyScheduler, HiQMainEngine

from mpi4py import MPI

def adiabatic_simulation():
    """The function you need to modify.
    Returns:
        real_energy(float):
            The final ideally continously evolved energy.
        simulated_energy(float):
            The final energy simulated by your model.
    """
    simulated_energy = 0
    real_energy = 0
    return simulated_energy, real_energy

if __name__ == "__main__":

    # use projectq simulator
    #eng = MainEngine()

    # use hiq simulator
    backend = SimulatorMPI(gate_fusion=True)

    cache_depth = 10
    rule_set = DecompositionRuleSet(modules=[projectq.setups.decompositions])
    engines = [TagRemover()
               , LocalOptimizer(cache_depth)
               , AutoReplacer(rule_set)
               , TagRemover()
               , LocalOptimizer(cache_depth)
               , GreedyScheduler()
               ]

    # make the compiler and run the circuit on the simulator backend
    eng = HiQMainEngine(backend, engines)
```

```
51
52      qureg = eng.allocate_qureg(5)
53
54      simulated_energy, real_energy = adiabatic_simulation()
55
56      simulated_error = simulated_energy - real_energy
57
58      All(Measure) | qureg
59
60      print(simulated_error)
```