
My energy supplier pal

There are many energy suppliers and their rates vary depending on consumption. The requested program should help a customer choose the plan that is best for them, by:

- Evaluating plans based on the consumed energy
- Calculating how much energy is used based on customer spend

Kilowatt-hour (kWh) is the unit of energy used for recording energy consumption.

VAT for energy is 5%.

Plans contain a set of prices that describe how much a customer will be charged for each kWh of energy they use.

Plans may include a daily standing charge; it is expressed in pence exclusive of VAT and is applied regardless of consumption.

Plans may have more than one price. When a plan does have multiple prices, the last one will always be with no threshold and the rest will have a threshold.

Thresholds indicate the amount of energy (up to and including) that may be consumed at that rate annually. Rates without a threshold have no limit.

Rates are stated in pence and are exclusive of VAT.

In the example json below, the first 100kWh will be charged at 15p/kWh, the next 150kWh will be charged at 12.5p/kWh and all subsequent consumption will be charged at 10p/kWh.

```
{
  "supplier_name": "energySupplier",
  "plan_name": "planOne",
  "prices": [
    {
      "rate": 15,
      "threshold": 100
    },
    {
      "rate": 12.5,
      "threshold": 150
    },
    {
      "rate": 10
    }
  ],
  "standing_charge": 5
}
```

Software Requirements

Input

Plans are imported from a json file. The schema is:

```
{
  "$schema": "http://json-schema.org/myenergypal/schema#",
  "description": "Schema for validating a Plan instance.",
  "id": "https://www.mizuho-emea.com/myenergypalschema.json",
  "properties": {
    "plan_name": {
      "description": "Unique Plan Name",
      "id": "/properties/plan_name",
      "title": "Plan Name",
      "type": "string"
    },
    "prices": {
      "additionalItems": false,
      "id": "/properties/prices",
      "items": {
        "id": "/properties/prices/items",
        "properties": {
          "rate": {
            "description": "Rate expressed in pence; exclusive of VAT",
            "id": "/properties/prices/items/properties/rate",
            "title": "Rate",
            "type": "number"
          },
          "threshold": {
            "description": "Energy to be consumed at this rate annually",
            "id": "/properties/prices/items/properties/threshold",
            "title": "Threshold",
            "type": "integer"
          }
        }
      },
      "type": "object"
    },
    "standing_charge": {
      "description": "Daily charge expressed in pence; exclusive of VAT",
      "id": "/properties/standing_charge",
      "title": "Standing Charge",
      "type": "integer"
    },
    "supplier_name": {
      "description": "Supplier Name",
      "id": "/properties/supplier_name",
      "title": "Supplier Name",
      "type": "string"
    }
  },
  "title": "Root JSON Schema.",
  "type": "array"
}
```

Functionality

Annual Cost Calculation:

For a given annual kWh consumption, it calculates the annual total cost in pounds (rounded to 2 decimal places), inclusive of VAT, for all plans available on the market in ascending order (i.e. cheaper first).

For each plan, the following information should be provided:

Supplier Name	Plan Name	Total Cost
---------------	-----------	------------

Console Application

A user can pass three different commands to the program, namely:

- `input <filename>`

E.g. `input C:\Users\Public\TestFolder\plans.json`

, which will load the plans for further usage.

- `annual_cost <annual_consumption>`

E.g. `annual_cost 5000`

, which will call the 'Annual Cost Calculation' function and will print to stdout in comma separated format the Supplier Name, Plan Name and Total Cost.

- `exit`

, which will terminate the program.

Output

Please write your solution in C# 6.0 (.NET 4.6 and VS2015) or latest. Your program should both produce the right output and also be well written. It will be verified with the provided inputs and plans (see attachment) and is considered correct if it produces the expected output.

Submission

Your program should be submitted in a zip file, containing the sln and just the relevant code files (i.e. not compiled code). Package dependencies should be resolved with Nuget.

When you are invited for the face-to-face interview we will pair with you to extend your program.

We appreciate people have day jobs and other commitments, so please take your time to solve the task fully. Expected estimated effort is 2-3 hours. Please let us know if you cannot submit your answer within 5 calendar days of receiving the assignment.

Please do not share your solution online in personal blogs, github, etc.

Thank you.

Attachments

plans.json

```
[
  {
    "supplier_name": "energyOne", "plan_name": "planOne",
    "prices": [
      {
        "rate": 13.5, "threshold": 100
      },
      {
        "rate": 10
      }
    ]
  },
  {
    "supplier_name": "energyTwo", "plan_name": "planTwo",
    "prices": [
      {
        "rate": 12.5, "threshold": 300
      },
      {
        "rate": 11
      }
    ]
  },
  {
    "supplier_name": "energyThree", "plan_name": "planThree",
    "prices": [
      {
        "rate": 14.5, "threshold": 250
      },
      {
        "rate": 10.1, "threshold": 200
      },
      {
        "rate": 9
      }
    ]
  },
  {
    "supplier_name": "energyFour", "plan_name": "planFour",
    "prices": [
      {
        "rate": 9
      }
    ],
    "standing_charge": 7
  }
]
```

Commands

```
input plans.json  
annual_cost 1000  
annual_cost 2000  
exit
```

Expected output

```
energyOne,planOne,108.68  
energyThree,planThree,111.25  
energyTwo,planTwo,120.22  
energyFour,planFour,121.33  
energyThree,planThree,205.75  
energyOne,planOne,213.68  
energyFour,planFour,215.83  
energyTwo,planTwo,235.72
```