Python Clearcode intern summer 2019 task

Dark times have come. Article 13 has been passed almost 2 years ago. Memes are illegal. People use USB sticks to store and sell them for caps. Every meme is identified by a size, given in MiB, and its market price. xXxDankScavengerxXx sells memes as his way of earning a living. Help him by writing function `calculate(usb_size, memes)` that calculates the best set of memes, so that he can sell the USB stick for the highest price.

- `usb_size: int`
  a number describing the capacity of the USB stick in GiB - e.g. `1` means a USB with 1 GiB capacity.
- `memes: List[Tuple[str, int, int]]`
  is a list of 3-element tuples, each with the name, size in MiB, and price in caps of a meme:
  e.g. `[('dolan.png', 126, 5), ('expanding_brain.jpeg', 421, 10)]`
  which means that the xXxDankScavengerxXx has two memes to sell: `'dolan.png'` which is 126 MiB and can be sold for 5 caps and `'expanding_brain.jpeg'` which is 421 MiB and can be sold for 10 caps.

The function should return a tuple with the first element being the total value of all memes on the USB stick, and the second being the set of names of the memes that should be copied onto the USB stick to maximize its value.
e.g. `(15, {'dolan.png', 'expanding_brain.jpeg'})`
Means that the USB stick can be sold for 15 caps if it contains `'dolan.png'` and `'expanding_brain.jpeg'`.

Example:

```
usb_size = 1
memes = [
      ('rollsafe.jpg', 205, 6),
      ('sad_pepe_compilation.gif', 410, 10),
      ('yodeling_kid.avi', 605, 12)
]

calculate(usb_size, memes)
```

should return

```
(22, {'sad_pepe_compilation.gif', 'yodeling_kid.avi'})
```

Requirements:

- Each meme can be used at most once (who would like to see a repost anyway).
- The function should be written in Python 3 (preferably 3.7).
- The function must take exactly 2 arguments, `usb_size` and `memes`
- If your submission doesn't run because of any errors (syntax error, import error, etc.) we will neither check nor grade it.
- If you use any external packages, please provide a list of them in a file named `requirements.txt` (we will not try to guess which package you are trying to import), so we can install them before running our tests.
- Name the file that we can import the function `calculate` from "`main.py`", so that we know where to find your solution. If your solution uses multiple files, make sure that we can import `calculate` from `main.py` with `from main import calculate`.
- Send us your solution preferably by linking to a git repository (github, bitbucket, gitlab, etc.), or as a .zip, .7z, .xz, .lzma, or .tar.gz archive (please don't use rar, we Linux users don't like it very much).
- We will check the computational and memory complexity of your solution as well as quality and readability of your code and if your submission gives correct answers to our test cases of course.
- We expect you to provide us with some documentation about your solution (e.g. `README` file, docstrings).

Good luck!