Programowanie w Pythonie - test praktyczny

Zadanie1 (10pkt): Napisz skrypt w języku Python który:

- otwiera plik tekstowy zawierający pewną notatkę historyczną (wydarzenia opatrzone datami - latami);

- zapisuje do listy lata z tekstu; powinieneś otrzymać listę [1000, 1200, 1400]

- wyznacza w sposób automatyczny różnicę w latach pomiędzy najwcześniejszym a najpóźniejszym wydarzeniem opisanym w tej notatce.

Zadanie2 (8pkt): Napisz funkcję, która przyjmuje nieznaną ilość stringów w postaci 'Imie Nazwisko' i zwraca listę tych stringów posortowaną według nazwisk.

Przykładowo: `program('Adam Kampinos', 'Angela Bobka', 'Jan Niezbedny')` zwróci `['Angela Bobka', 'Adam Kampinos', 'Jan Niezbedny']`

Zadanie3 (10pkt): Klasa `Informatyk` dziedziczy po klasie `Pracownik`. Każda instancja tej klasy posiada imię, nazwisko, średnią liczbę przepracowanych godzin w miesiącu oraz stawkę godzinową. Zdefiniuj klasę `Pracownik` - klasę rodzic dla klasy `Informatyk` tak aby możliwe było zdefiniowanie instancji *t*, a następnie wywołanie kolejnych metod (patrz niżej). Nie należy modyfikować klasy `Informatyk`!

```
class Pracownik:
    pass

class Informatyk(Pracownik):
    def __init__(self, imie, nazwisko, godziny, stawka):
        super().__init__(imie, nazwisko, godziny)
        self.stawka = stawka
    def ile_miesiecznie(self):
        return self.godziny*self.stawka
```

```
t = Informatyk("Tomek", "Misztal", 200, 30)
print(t) #metoda print zwraca imie
```

```
Tomek
```

```
print(t.przedstaw_sie()) #metoda przedstaw_się zwraca napis jak niżej
```

```
Witaj, nazywam sie Tomek Misztal.
```

```
print(t.godziny_miesiace(2)) #meoda godziny_miesiące zwraca liczbę godzin
przepracowanych w ciagu n miesiecy, tutaj n=2.
```

```
400
```

```
print(t.ile_miesiecznie()) #metoda zwraca miesięczne wynagrodzenie,
wynagrodzenie = stawka godzinowa * liczba godzin
```

```
6000
```

Zadanie4 (7pkt): Napisz funkcję, która przyjmuje jako argument listę liczb całkowitych dodatnich i która zwraca listę zagnieżdżonych list z tymi liczbami. Stopień zagnieżdżenia określony jest na podstawie wartości elementów z listy.

`[1,1,3,2]`

`zwróci`

`[[1],[1],[[[3]]],[[2]]]`