

Lisaülesanne 8.1

a) Asendusšifrite murdmiseks on hea vahend märkide esinemissageduste võrdlus märkide esinemissagedustega oodatavas keeles. Definiirida funktsioon `sagedused(failinimi)`, mis tagastab failinimega määratud tekstifailis sisalduvate märkide sagedused sõnastikuna kujul `{märk1: sagedus1, märk2: sagedus2, ...}`.

Näiteks `sagedused("kodeeritud_yl.txt")` tagastab:

```
{'ä': 252, '...': 1, ' ': 2666, '•': 6, ',': 339, '.': 145, '(': 9,
')': 83, '1': 28, '2': 44, ':': 142, '\n': 12, 'd': 729, 'e':
2295, 'f': 251, 'g': 379, 'a': 1867, 'b': 135, 'c': 23, 'l': 805,
'm': 1015, 'n': 233, 'o': 717, 'h': 34, 'i': 514, 'j': 331, 'k':
1211, 't': 1569, 'u': 2279, 'v': 592, 'p': 144, 'r': 641, 's':
1051, 'ü': 78, '}' : 1, 'x': 11, '{': 6, 'D': 17, 'E': 14, 'F': 3,
'G': 19, 'ö': 56, 'L': 57, 'N': 29, 'H': 16, 'K': 73, 'T': 28,
'V': 15, 'S': 19, '_': 9}
```

b) Trükkida sellise sõnastiku (märk, sagedus) paarid välja sorditult, suurimast sagedusest väikseimani.

Siin on kasulik teada, et funktsioon `sorted(jada, key=funktsioon, reverse=True/False)` omab valikulisi parameetreid. Parameeter `key` peab olema üheelemendiline funktsioon, mis teisendab jada elemendi võtmeks (`key`), mille alusel sortimine toimub.

Näiteks, kui tahame sortida arve, mitte nende suuruse, vaid viimase numbr järgi, siis võime tuua sisse abifunktsiooni, mis tagastab arvu viimase numbr (soovitud sortimise võtme):

```
def viimane_number(n):
    return int(str(n)[-1])
```

Väljakutse `sorted([108, 302, 103, 9], key=viimane_number)` tagastab siis `[302, 103, 108, 9]`.

Parameeter `reverse` määrab kahaneva või kasvava sortimise järjekorra.