



Osnove 4

Python

Python urejevalnik se nahaja na strani:

<https://www.w3resource.com/python-exercises/python-basic-exercises.php#EDITOR>

Naloge

1. **Kovanci** Na razpolago imamo evrske kovance in bankovce za vredosti: 1, 2, 5, 10, 20, 50, 100, 200 evrov. Ugotovi na koliko načinov lahko plačamo račun za n evrov, kjer je n celo število.

Primeri:

44 --> 27; 306 --> 1802;

2. **Permutacije** Izpišite vse permutacije (razporedbe) črk danega niza.

Primer:

'ABC' --> ['ABC', 'ACB', 'BAC', 'BCA', 'CAB', 'CBA']

3. **Oklepaji** Napišite program, ki nam bi pomagal ugotoviti napako pri grupiranju v aritmetičnem izrazu, kjer se uporabljajo tri vrste oklepajev. Okrogli (), oglati [] in zaviti {}. Ostale sintaktične napake pri pisanju aritmetičnega izraza nas ne zanimajo. Tudi zaporedje uporabe oklepajev ni pomembno.

Primeri:

{2+4*(1-3)+(2-4)*{2-3*[8+3*(3+5)]}} --> Pravilno

{2+4*(1-3)+(2-4)*2-3*[8+3*(3+5)]}} --> Napačno

4. **Delitelji** Poišči vse delitelje danega števila.

Primer:

144 --> [144, 72, 48, 36, 24, 18, 16, 12, 9, 8, 6, 4, 3, 2, 1]

Rešitve 4

Kovanci

```
import functools, time

coins = [1, 2, 5, 10, 20, 50, 100, 200]
def nways(n):
    ways = [1] + [0] * coins[-1]
    for coin in coins:
        for i in range(coin, n + 1):
            ways[i] += ways[i - coin]
    return (ways[n])

if __name__ == '__main__':
    n = int(input('n -> '))
    print("Načinov plačila = %d" % nways(n))
```

Permutacije

```
#!/usr/bin/env python3

def permute(word):
    res = []
    if len(word) == 1:
        return [word]
    else:
        for i in range(len(word)):
            sub_permute = permute(word[:i]+word[i+1:])
            for item in sub_permute:
                res.append(word[i]+item)
    return res

if __name__ == '__main__':
    print(permute('ABCD'))
```

Oklepaji

```
#!/usr/bin/env python3

bb = {'(': ')', '[': ']', '{': '}'

def parse(line):
    braces = []
    i = 0
    for ch in line:
        i += 1
        if ch in bb.keys():
            braces.append(ch)
        elif ch in bb.values():
            try:
                if bb[braces.pop()] != ch:
                    return (False, i)
            except:
                return (False, i)

    return (braces == [], i, braces)

if __name__ == '__main__':
    line = input('Vpiši izraz: ')
    print(parse(line))
```

Faktorji

```
#!/usr/bin/env python3

def sfactors(n):
    k = 1
    while k * k < n:
        if n % k == 0:
            yield [k, n // k]
        k += 1
    if k * k == n:
        yield [k]

if __name__ == '__main__':
    factors = []
    for factor in sfactors(144):
        factors += factor
    print(sorted(factors, reverse=True))
```