PYTHON

Osnove 5

Python urejevalnik se nahaja na strani:
`https://www.w3resource.com/python-exercises/python-basic-exercises.php#EDITOR`
Naloge Kattis:`https://github.com/minidomo/Kattis`
Šolska liga programiranje:
`https://www.zotks.si/programiranje/novice/druga-tekma-%C5%A1olske-lige-programiranja`

1. **Magično število 6147** Izberi štirištevilčno število n, kjer niso vse števke enake. Napiši funkcijo $f(n)$, ki vrne razliko med največjim številom, ki ga dobimo s permutacijo števk tega števila in najmanjšim takim številom. V zanki ponavljaj $n = f(n)$ toliko časa, da bo število na vhodu enako številu na izhodu.

   Na vhodu preveri, da je število štirištevilčno, če ni vrni vrednost -1. Nato še preveri ali so vse štiri števke enake, če so vrni vrednost 0.

   Primer:

   ```
   n-> 2312: 1998 8082 8532 6174 6174
   ```

2. **Dan v letu 2022** 1. januar v letu 2022 je bila sobota.
   Iz standardnega vhoda preberi število $1 \leq n \leq 365$. Izpiši datum tega dne v letu.
   Primer:

   ```
   n-> 256: torek 13. september
   ```

   Za ostale dneve lahko preverite na naslovu:
   `https://www.dayoftheweek.org/`

3. **Palindromsko število** Palindromsko število je število, ki se prebere enako naprej in nazaj. Na primer 1234321. Za dano število bomo izbrali njemu prirejeno palindromsko število.

   a) Preberemo število n iz standardnega vhoda.

   b) Izračunamo vsoto števila n in števila, ki ima števke v obratnem vrstnem redu.

   c) Če vsota ni palindromsko število, potem ponovimo 3b.

   d) Ponavljamo dokler ne dobimo vsote, ki je palindromsko število.

   Je mogoče, da za poljubno število gornji algoritem pripelje do palindromskega števila? Problem ni rešen. Za določena števila je število korakov lahko zelo veliko, če že ne neskončno.

   Na primer za števila 196, 295 in 394 ne vemo ali se postopek konča s palindromskim številom.
   Primer:

```
n-> 3462: 6105 11121 23232
```

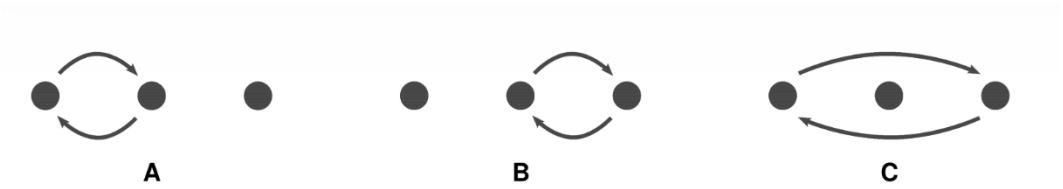
4. **Igra "tri čaše"** Na mizi so tri čaše. V igri sta dva igralca, Čaše niso prozorne in obrnjene so z dnom navzgor. Oštevilčimo čaše z 0, 1, 2, tako da bodo oznake usklajene z indeksiranjem seznamov v sistemu `Python`. Pod eno čašo položimo kroglico. Prvi igralec meša čaše, medtem ko drugi igralec budno opazuje mešanje čaš. Na koncu mora povedati, pod katero čašo se nahaja kroglica.

   Začetno stanje opišemo s seznamom `stanje = [1, 0, 0]`. Čaša s kroglico se nahaja na skrajni levi `stanje[0] == 1`. Dovoljeni premiki čaš so prikazani na sliki.

   a) Premik `A` je zamenjava `stanje[0] <--> stanje[1]`,

   b) premik `B` je zamenjava `stanje[1] <--> stanje[2]` in

   c) premik `C` je zamenjava `stanje[2] <--> stanje[0]`.



   Iz standardnega vhoda preberite poljubno dolg niz z znaki `A, B, C` in na standardni izhod izpišite stanje na koncu.

   Primera:

```
AB         --> [0, 0, 1]
CBABCACCC  --> [1, 0, 0]
```

   Če na poznate igre si poglejte na naslednji povezavi:
   `https://www.youtube.com/shorts/V8NweYOHzeg`

5. **Bruci**

```
komb = '''100 101 102 103 112
100 110 111 101 102
102 110 101 111 101
103 102 101 112 100
100 110 111 101 102
100 102 101 112 103
103 112 102 111 100
100 110 111 112 109
103 111 102 112 100
100 112 103 111 102
111 105 106 109 100
103 111 102 112 100
102 103 109 110 111'''
```

Pri vpisu na fakulteto je vsak bruc moral izbrati pet različnih predmetov, ki jih bo poslušal v prvem letniku. Predmeti so označeni s številkami od 100 do 112. Na primer v tabeli `komb` so podatki za 13 brucev. Opozorimo naj, da so podatki v tabeli pomešani. Tabele so lahko poljubno dolge.

Primer

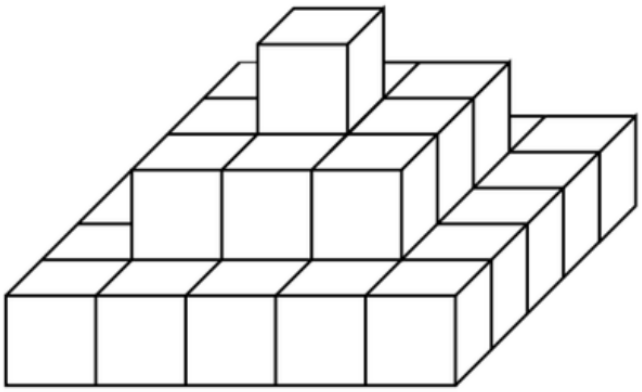
```
1. '100 102 103 111 112': 4
2. '100': 11, '102': 11
```

a) Iščemo najbolj popularne kombinacije predmetov.

b) Katere predmete so bruci izbrali največkrat.

Program napišite tako, da bo deloval za poljubno dolgo tabelo.

6. **Piramida**



Piramida na sliki je polna. Znotraj nje ne manjkajo bloki.

Vaša naloga je, da za dano je število blokov, ki jih imamo na razpolago, izračunate največ kako visoko piramido je mogoče narediti, če jo gradimo po pravilih, kot je prikazano na sliki.

Primer

```
1. 50 --> 3
2. 83 --> 3
3. 84 --> 4
```

Rešitve 9

Magično število

```python
#!/usr/bin/env python3
#https://www.youtube.com/watch?v=pDXek06Bde4&t=39s

def n6174(n):
    l = len(n)
    mi = "".join(sorted(n))
    mx = "".join(sorted(n, reverse=True))
    n = str(int(mx) - int(mi))
    return "0"*(l - len(n)) + n

if __name__ == '__main__':
    n = input(' n-> ')
    if len(n) == 4:
        m = n6174(n)
        print(m)
        while m != n:
            n = m
            m = n6174(n)
            print(m)
```

Dan v letu 2022

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

teden = ['nedelja', 'ponedeljek', 'torek', 'sreda', 'četrtek', 'petek', 'sobota']

meseci = {'januar': 31, 'februar': 28, 'marec': 31, 'april': 30, 'maj': 31,
          'junij':30, 'julij':31, 'avgust':31, 'september': 30,
          'oktober': 31,  'november': 30,  'december': 31}

def prvi_dan(leto):
    y = leto % 100 - 1; c = leto // 100;
    return (29 - 2*c + y + y//4 + c//4) % 7


def datum(n, leto):
    if not (leto % 4):
        meseci['februar'] += 1
    dan = teden[(prvi_dan(leto) + n - 1) % 7]
    d = 0
    for m in meseci:
        d += meseci[m]
        if d >= n:
            mesec = m
            return dan + ' ' + str(meseci[mesec] - (d - n)) + '. '+ mesec

if __name__ == '__main__':
    n, leto = [int(x) for x in input('n leto ').split()]
    print(datum(n, leto))
    print(prvi_dan(leto))
```

Palindrom

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

def palindrom(n):
    m = n[::-1]
    return str(int(n) + int(m))

if __name__ == '__main__':
    n = input('n -> ')
    m = palindrom(n)
    while m != n[::-1]:
        n = palindrom(m)
        m = n
        print(n)
```

Tri čaše

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

def abc(st):
    def a(res):
        res[0], res[1] = res[1], res[0]
        return res
    def b(res):
        res[1], res[2] = res[2], res[1]
        return res
    def c(res):
        res[0], res[2] = res[2], res[0]
        return res

    func = {'A': a, 'B': b, 'C': c}
    res = [1, 0, 0]
    for x in st:
        func[x](res)
    return res

if __name__ == '__main__':
    st = input('abc -> ')
    print(abc(st))
```

Bruci

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

comb = '''100 101 102 103 112
100 110 111 101 102
102 110 101 111 101
103 102 101 112 100
100 110 111 101 102
100 102 101 112 103
103 112 102 111 100
100 110 111 112 109
103 111 102 112 100
100 112 103 111 102
111 105 106 109 100
103 111 102 112 100
102 103 109 110 111'''

def popular(comb):
    line = [" ".join(sorted(x.split())) for x in comb.split("\n")]
    keys = set(line)
    freq = dict(zip(keys, [0 for i in range(len(keys))]))
    for x in line:
        freq[x] += 1
    return freq, max(freq.values())

def maximal(comb):
    line = [" ".join(sorted(x.split())) for x in comb.split("\n")]
    extl = []
    [extl.extend(x.split()) for x in line]
    keys = set(extl)
    freq = dict(zip(keys,[0 for _ in keys]))
    for x in extl:
        freq[x] += 1
    return freq

print(maximal(comb))
```

Piramida

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

def visina(n):
    s = 0
    m = 1
    i = 0
    while s <= n:
        print(s, i, m)
        i += 1
        s += m**2
        m += 2
    return i - 1

if __name__ == '__main__':
    n = int(input('n -> '))
    print(visina(n))
```