

Vaje pri predmetu Programiranje 1

Teden 3: Osnovni konstrukti III

Piramida

Naloga

Napišite program, ki prebere pozitivno celo število n , nato pa nariše »piramido« zvezdic višine n , kot jo prikazujejo primeri v nadaljevanju.

Vhod

Na vhodu je podano celo število $n \in [1, 100]$.

Izhod

Na izhod izpišite »piramido« po zgledu sledečih primerov. Ne izpisujte odvečnih presledkov in praznih vrstic!

Testni primer 1

Vhod:

4

Izhod:

```
*  
***  
*****  
******
```

Izhod s prikazanimi presledki:

```
  *  
  ***  
  *****  
  ******
```

Testni primer 2

Vhod:

1

Izhod:

*

Kleptomanka Anka

Naloga

Kleptomanka Anka¹ bi rada ukradla kolo, ki je zaščiteno s številčno ključavnico. Ključavnica ima tri številska mesta ($a-b-c$). Nekdo Anki prisepne, da je prva števka liha, druga večja od m , tretja pa deljiva z d . Napišite program, ki prebere števili m in d ter izpiše vse možne kombinacije števk. Na koncu naj izpiše število vseh kombinacij.

Vhod

Na vhodu sta podani celi števili $m \in [0, 8]$ in $d \in [1, 9]$, ločeni s presledkom.

Izhod

Vsako kombinacijo izpišite v svojo vrstico, in sicer v sledeči obliki:

$a-b-c$

Kombinacije izpišite v leksikografskem vrstnem redu: najprej naraščajoče po prvi števki, nato po drugi in nazadnje po tretji. Po izpisu vseh kombinacij izpišite še število kombinacij.

Testni primer

Vhod:

7 3

Izhod:

1-8-0
1-8-3
1-8-6
1-8-9
1-9-0
1-9-3
1-9-6
1-9-9
3-8-0
3-8-3
3-8-6
3-8-9
3-9-0
3-9-3

¹Po skladbi *Anka Kleptomanka* Romane Krajnčan

3-9-6
3-9-9
5-8-0
5-8-3
5-8-6
5-8-9
5-9-0
5-9-3
5-9-6
5-9-9
7-8-0
7-8-3
7-8-6
7-8-9
7-9-0
7-9-3
7-9-6
7-9-9
9-8-0
9-8-3
9-8-6
9-8-9
9-9-0
9-9-3
9-9-6
9-9-9
40

Zgoščenke

Naloga

Svoje ključne datoteke želimo shraniti na n praznih zgoščenk, od katerih lahko vsaka hrani po k enot podatkov. Datoteke (nobena ne vsebuje več kot k enot podatkov) obravnavamo po vrsti. Najprej shranjujemo datoteke na prvo zgoščenko. Ko nam zmanjka prostora za zapis trenutne datoteke, preidemo na drugo zgoščenko; ko nam zmanjka prostora še na tej, preidemo na tretjo itd. Na predhodne zgoščenke se nikoli ne vračamo, četudi bi bilo na kateri od njih še dovolj prostora za trenutno datoteko.

Napišite program, ki najprej prebere števili n in k , nato pa zaporedoma bere velikosti datotek. Program naj za vsako datoteko izpiše, na katero zgoščenko (prvo, drugo, tretjo, ...) se shrani in koliko podatkov je trenutno shranjenih na tej zgoščenki. Program naj se zaključi, ko zmanjka vhoda ali pa zgoščenk.

Vhod

V prvi vrstici sta podani celi števili $n \in [1, 10^4]$ in $k \in [1, 10^9]$, ločeni s presledkom, nato pa sledi zaporedje celih števil z intervala $[1, k]$, ki podajajo velikosti posameznih datotek. Vsaka velikost je zapisana v svoji vrstici.

Izhod

Za vsako uspešno shranjeno datoteko izpišite po eno vrstico v sledeči obliki:

$X \text{ EP} \rightarrow \text{zgoscenka } Y \text{ EP}$

Pri tem je X velikost tekoče datoteke, Y številka zgoščenke, kamor se shrani tekoča datoteka, Z pa trenutna količina podatkov na tej zgoščenki.

Testni primer 1

Vhod:

```
4 700
500
200
300
600
50
```

Izhod:

```
500 EP -> zgoscenka 1 (500 EP)
200 EP -> zgoscenka 1 (700 EP)
300 EP -> zgoscenka 2 (300 EP)
600 EP -> zgoscenka 3 (600 EP)
50 EP -> zgoscenka 3 (650 EP)
```

Testni primer 2

Vhod:

```
4 700
500
200
300
600
50
100
400
500
550
```

Izhod:

```
500 EP -> zgoscenka 1 (500 EP)
200 EP -> zgoscenka 1 (700 EP)
300 EP -> zgoscenka 2 (300 EP)
600 EP -> zgoscenka 3 (600 EP)
50 EP -> zgoscenka 3 (650 EP)
100 EP -> zgoscenka 4 (100 EP)
400 EP -> zgoscenka 4 (500 EP)
```

V tem primeru zadnjih dveh datotek ne moremo več shraniti.