

## Dodatne naloge za 3. teden predavanj

### Splošna navodila

Pri vseh nalogah v tem sklopu lahko svoje rešitve preverite z množico vhodnih in pripadajočih izhodnih datotek.

### 1 Piramida števil

#### Naloga

Napišite program, ki prebere število  $n$  in nariše »piramido« števil višine  $n$ , kot jo prikazujeta primera v nadaljevanju.

#### Vhod

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

#### Izhod

Izpišite »piramido« po zgledu sledečih primerov. Ne izpisujte odvečnih presledkov in praznih vrstic.

#### Testni primer 1

Vhod:

5

Izhod:

```
  1
 234
34567
4567890
567890123
```

Izhod s prikazanimi presledki:

```
 1
 234
34567
4567890
567890123
```

## Testni primer 2

Vhod:

```
11
```

Izhod:

```
1
234
34567
4567890
567890123
67890123456
7890123456789
890123456789012
90123456789012345
0123456789012345678
123456789012345678901
```

## 2 Igorjevi bloki

### Naloga

Napišite program, ki prebere tri enomestna števila in nariše vzorec, kot ga prikazujeta primera v nadaljevanju.

### Vhod

Na vhodu so podana tri cela števila z intervala  $[1, 9]$ , ločena s presledkom.

### Izhod

Izpišite vzorec po zgledu sledečih primerov. Ne izpisujte odvečnih presledkov in praznih vrstic.

## Testni primer 1

Vhod:

```
3 7 4
```

Izhod:

```
333 7777777 4444
333 7777777 4444
333 7777777 4444
    7777777 4444
    7777777
    7777777
    7777777
```

Izhod s prikazanimi presledki:

```
333_7777777_4444
333_7777777_4444
333_7777777_4444
_7777777_4444
_7777777
_7777777
_7777777
_7777777
```

## Testni primer 2

Vhod:

```
2 2 3
```

Izhod:

```
22 22 333
22 22 333
    333
```

## 3 Šahovnica

### Naloga

Napišite program, ki prebere števila  $v$ ,  $s$  in  $d$  in nariše vzorec v obliki šahovnice z  $v$  vrsticami in  $s$  stolpci, pri čemer ima vsako polje obliko kvadrata velikosti  $d \times d$ . Šahovnica naj bo tudi obrobljena. Zgledujte se po primerih v nadaljevanju.

### Vhod

Na vhodu so podana cela števila  $v$ ,  $s$  in  $d$  z intervala  $[1, 20]$ . Med seboj so ločena s presledkom.

### Izhod

Izpišite vzorec po zgledu sledečih primerov. Ne izpisujte odvečnih presledkov in praznih vrstic.

## Testni primer 1

Vhod:

```
3 4 5
```

Izhod:

```

+ - - - - - - - - - - - - - - - +
|           * * * * *           * * * * * |
|           * * * * *           * * * * * |
|           * * * * *           * * * * * |
|           * * * * *           * * * * * |
|           * * * * *           * * * * * |
| * * * * *           * * * * * |
| * * * * *           * * * * * |
| * * * * *           * * * * * |
| * * * * *           * * * * * |
| * * * * *           * * * * * |
|           * * * * *           * * * * * |
|           * * * * *           * * * * * |
|           * * * * *           * * * * * |
|           * * * * *           * * * * * |
|           * * * * *           * * * * * |
+ - - - - - - - - - - - - - - - +

```

## Testni primer 2

Vhod:

6 5 2

Izhod:

```

+ - - - - - - - - - - +
|   * *   * *   |
|   * *   * *   |
| * *   * *   * * |
| * *   * *   * * |
|   * *   * *   |
|   * *   * *   |
| * *   * *   * * |
| * *   * *   * * |
|   * *   * *   |
|   * *   * *   |
| * *   * *   * * |
| * *   * *   * * |
+ - - - - - - - - - - +

```

Izhod s prikazanimi presledki:

```

+ _ _ _ _ _ _ _ _ _ _ +
| _ _ _ _ _ * _ * _ _ _ _ * _ * _ _ _ _ |
| _ _ _ _ _ * _ * _ _ _ _ * _ * _ _ _ _ |
| _ * _ * _ _ _ _ * _ * _ _ _ _ * _ * _ |
| _ * _ * _ _ _ _ * _ * _ _ _ _ * _ * _ |
| _ _ _ _ _ * _ * _ _ _ _ * _ * _ _ _ _ |
| _ _ _ _ _ * _ * _ _ _ _ * _ * _ _ _ _ |
| _ * _ * _ _ _ _ * _ * _ _ _ _ * _ * _ |
| _ * _ * _ _ _ _ * _ * _ _ _ _ * _ * _ |
+ - - - - - - - - - - +

```

```
|uuuuu*uuuuu*uuuuu*|
|uuuuu*uuuuu*uuuuu*|
|u*uuuuu*uuuuu*uuuuu|
|u*uuuuu*uuuuu*uuuuu|
+u-u-u-u-u-u-u-u-u+
```

## 4 Anžetove ledene sveče

### Naloga

Napišite program, ki prebere število  $n$  in nariše vzorec višine  $n$ , kot ga prikazujejo primeri v nadaljevanju.

### Vhod

Na vhodu je podano celo število  $n \in [2, 20]$ .

### Izhod

Izpišite vzorec po zgledu sledečih primerov. Ne izpisujte odvečnih presledkov in praznih vrstic.

#### Testni primer 1

Vhod:

5

Izhod:

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

#### Testni primer 2

Vhod:

6

Izhod:

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

### Testni primer 3

Vhod:

7

Izhod:

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

## 5 Metaprogram

### Naloga

S sledečo zanko po vrsti izpišemo vse velike črke angleške abecede od A do Z:

```
for (char c1 = 'A'; c1 <= 'Z'; c1++) {
    System.out.println(" " + c1);
}
```

Sedaj pa bi želeli po abecednem vrstnem redu izpisati vse nize (besede), sestavljene iz  $n$  velikih črk angleške abecede. Na primer, če je  $n = 2$ , bi želeli izpisati nize AA, AB, ..., AZ, BA, BB, ..., BZ, ..., ZA, ZB, ..., ZZ. Če je  $n = 3$ , bi želeli izpisati nize AAA, AAB, ..., ZZZ. Da bi lahko nalogo rešili za poljuben  $n$ , bi potrebovali nekoliko več znanja, kot ga imamo sedaj. Za fiksno  $n$  pa lahko nalogo rešimo s pomočjo  $n$  vgnezenih zank. Ker program seveda ne more vsebovati spremenljivega števila zank, boste napisali *metaprogram* — program, ki izpiše program, ki opisani problem reši s pomočjo  $n$  vgnezenih zank. Natančno se zgledujte po primeru, prikazanem v nadaljevanju.

### Vhod

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

### Izhod

Izpišite program po zgledu sledečega testnega primera. Natančno se držite števila presledkov med posameznimi elementi izpisanega programa. Ne izpisujte tabulatorjev, odvečnih presledkov in praznih vrstic.

### Testni primer 1

Vhod:

4

Izhod:

```
public class Nizi {
    public static void main(String[] args) {
        for (char c1 = 'A'; c1 <= 'Z'; c1++) {
            for (char c2 = 'A'; c2 <= 'Z'; c2++) {
                for (char c3 = 'A'; c3 <= 'Z'; c3++) {
                    for (char c4 = 'A'; c4 <= 'Z'; c4++) {
                        System.out.println("" + c1 + c2 + c3 + c4);
                    }
                }
            }
        }
    }
}
```

Izhod s prikazanimi presledki:

```
public class Nizi {
    public static void main(String[] args) {
        for (char c1 = 'A'; c1 <= 'Z'; c1++) {
            for (char c2 = 'A'; c2 <= 'Z'; c2++) {
                for (char c3 = 'A'; c3 <= 'Z'; c3++) {
                    for (char c4 = 'A'; c4 <= 'Z'; c4++) {
                        System.out.println("" + c1 + c2 + c3 + c4);
                    }
                }
            }
        }
    }
}
```

## Napotek

Če želite *izpisati* enojne ali dvojne navednice s pomočjo ukazov `System.out.print*`, uporabite zaporedje `"` oz. `'`. Na primer, ukaz

```
System.out.println("Znak \'a\' nastopa v nizu \"miza\".");
```

izpiše besedilo

```
Znak 'a' nastopa v nizu "miza".
```

## 6 Razbijanje števil

### Naloga

Napišite program, ki prebere števili  $n$  in  $m$  in po vrsti izpiše posamezne dele (zaporedja števk) števila  $n$ , pri čemer je dolžina posameznega dela določena s pripadajočo števkou v številu  $m$ . Dolžina prvega dela je tako enaka prvi števkou števila  $m$ , dolžina drugega dela je enaka drugi števkou števila  $m$  itd.

Nalogo rešite zgolj z operacijami nad celimi števili. Uporaba realnoštevilskih operacij ter nizov, tabel ipd. ni dovoljena.

### Vhod

Na vhodu sta podani celi števili  $n \in [1, 10^{18}]$  in  $m \in [1, 10^{18}]$ , ločeni s presledkom. Vsota števk števila  $m$  je enaka številu števk števila  $n$ . Števili  $n$  in  $m$  ne vsebujeta nobene ničle.

### Izhod

Izpišite toliko vrstic, kolikor je števk števila  $m$ . V prvi vrstici izpišite začetnih  $a_1$  števk števila  $n$  (pri čemer je  $a_1$  prva števka števila  $m$ ), v drugi sledečih  $a_2$  števk števila  $n$  (pri čemer je  $a_2$  druga števka števila  $m$ ) itd.

### Testni primer 1

Vhod:

```
3629831574865 2317
```

Izhod:

```
36
298
3
1574865
```