

APS 2 — druga domača naloga

Rok za oddajo: nedelja, 15. marca 2026

Karacubov algoritem

Naloga

Napišite program `karacuba.cpp`, ki prebere števila a , b in r , izpiše pa sled Karacubovega algoritma. Pri izvajanju algoritma rekurzijo »odrežite«, ko ima vsako od števil p in q , ki ju trenutno med seboj množite, večjemu r števku ali pa ko je vsaj eno od teh dveh števil enako 0. V takih primerih števil p in q ne zmnožite po Karacubovem postopku, ampak s klasičnim »papirnim« postopkom, ki teče v času $\Theta(n^2)$ (no, če je vsaj eno od števil p in q enako 0, nimate kaj množiti). Če ste pozabili, kako se števili zmnoži na papirju, povprašajte osnovnošolsko učiteljico iz sosednjega vhoda.

Vhodna števila nimajo vodilnih ničel; edina izjema je število 0, ki bi ga težko zapisali drugače. To naj velja tudi za vsa števila, ki jih izpisujete med izvajanjem postopka.

Prijateljski namig: morda vam bo lažje, če boste števila v tabeli (`vector<int>`, kajpada) hranili v obratnem vrstnem redu (najprej enice, nato desetice itd.).

Vhod

V prvi vrstici je podano število a , v drugi b , v tretji pa r . V vseh testnih primerih velja $a \in [0, 10^{10^4}]$, $b \in [0, 10^{10^4}]$ in $r \in [1, \max(\check{s}(a), \check{s}(b))]$, kjer $\check{s}(n)$ predstavlja število števk števila n . Testni primeri so pripravljeni tako, da število znakov, ki jih je treba izpisati, ne presega 10^7 .

V 50% testnih primerov velja $a \leq 10^9$ in $b \leq 10^9$.

Izhod

Po zgledu spodnjih primerov izpišite sled izvajanja Karacubovega algoritma pri podanem vhodu.

Testni primer 1

Vhod:

```
987
6543
4
```

Izhod:

```
987 * 6543
6457941
```

Testni primer 2

Vhod:

987
6543
3

Izhod:

987 * 6543
9 * 65
585
87 * 43
3741
96 * 108
10368
6457941

Testni primer 3

Vhod:

987
6543
2

Izhod:

987 * 6543
9 * 65
585
87 * 43
3741
96 * 108
0 * 1
0
96 * 8
768
96 * 9
864
10368
6457941

Testni primer 4

Vhod:

987
6543
1

Izhod:

987 * 6543
9 * 65
0 * 6
0
9 * 5

```

45
9 * 11
    0 * 1
    0
    9 * 1
    9
    9 * 2
    18
99
585
87 * 43
    8 * 4
    32
    7 * 3
    21
    15 * 7
        1 * 0
        0
        5 * 7
        35
        6 * 7
        42
105
3741
96 * 108
    0 * 1
    0
    96 * 8
        9 * 0
        0
        6 * 8
        48
        15 * 8
            1 * 0
            0
            5 * 8
            40
            6 * 8
            48
120
768
96 * 9
    9 * 0
    0
    6 * 9
    54
    15 * 9
        1 * 0
        0
        5 * 9
        45

```

```
        6 * 9
      54
    135
  864
10368
6457941
```

Oddaja naloge

Oddajte datoteko `karacuba.cpp`.

Povsem neobvezen dodatek

Če vas poleg algoritmike vsaj malo zanima tudi algoritemsko inženirstvo (kako algoritem učinkovito implementirati), vam priporočam sledeči poskus. Vzemite dovolj veliki (npr. 10 000-mestni) števili, odstranite vse izpise (ti upočasnjujejo delovanje) in poskusite ugotoviti, pri katerem r vaš algoritem deluje najhitreje. Z drugimi besedami: kdaj se vam splača rekurzijo odrezati in preiti na klasični postopek množenja?