

Darbības ar lieliem veseliem skaitļiem

`int` tipa datu maksimālā iespējamā vērtība ir nedaudz lielāka par 2 000 000 000.

```
int i_res = 2'000'000'000 + 2'000'000'000;  
std::cout << i_res; // = -294 967 296
```

```
std::cout << 2'000'000'000 + 2'000'000'000; // = -294 967 296
```

```
unsigned long long int ull_res = 2'000'000'000 + 2'000'000'000;  
std::cout << ull_res; // = 18 446 744 073 414 584 320
```

```
unsigned long long int ull_res = 2'000'000'000ULL + 2'000'000'000ULL;  
std::cout << ull_res; // = 4 000 000 000
```

```
std::cout << 2'000'000'000ULL + 2'000'000'000ULL; // = 4 000 000 000
```

Darbības ar lieliem veseliem skaitļiem

lai būtu vieglāk izmantot tipu ar garu nosaukumu, varat piešķirt tam īsu aizstāvjārdu

```
#include <iostream>

typedef unsigned long long int large; // 18 000 000 000 000 000 000

int main()
{
    large max = 100'000'000'000'000ULL;
    for (large i = 0; i < max; i += 1'000'000'000'000ULL)
        std::cout << i << "\n";
}
```

Mājas uzdevums: jaucējkode

Kāda vietne identificē lietotāju ar identifikatoru **ID**, par kuru kalpo veselais pirmskaitlis. Lai izvairītos no slepenas informācijas glabāšanas, administratori glabā datubāzē jaucējkodus (**HASH**), kas aprēķina kā **ID·KEY**, kur **KEY** ir cits pirmskaitlis, kas mazāks par **ID**. Lai verificētu lietotāja **ID**, vienkārši jāpārbauda, vai **HASH** ir vienmērīgi dalāms ar šo **ID**.

Hakeri ir nozadzis lietotāju datu bāzi, un viņiem ir jāiegūst lietotāju **ID** no jaucējkodiem. Uzrakstiet programmu, kas viņiem palīdzēs.

Piemēram:

HASH = **893 142 186 867 131** → ID = **123 456 461**

HASH = **1 234 567 813 222 222 867** → ID = **1 234 567 891**

Atkodēt jaucējkodus:

1 523 998 012 134 749

7 145 200 861 282 339

121 932 276 297 902 731

1 219 325 544 212 859 089