

```
root@879a04236d54:/work# pwd
/work
root@879a04236d54:/work# ls -l
total 8
-rw-r--r-- 1 root root 431 Aug 19 11:30 Dockerfile
-rw-r--r-- 1 root root   6 Aug 22 14:23 test.txt
root@879a04236d54:/work# cat test.txt
test
```

```
9      public class  hd1{
      Run | Debug
10      public static void main(String[] args) {
11          int a = 50000;
12          int b = 50000;
13          System.out.println(a*b);
14
15          float x = 1e20f;
16          float y = -1e20f;
17          float z = 3.14f;
18          System.out.println((x+y)+z);
19          System.out.println(x+(y+z));
20
21          double l = 1e20;
22          double m = -1e20;
23          double n = 3.14;
24          System.out.println((l+m)+n);
25          System.out.println(l+(m+n));
26
27
28      }
29  }
```

```
• (base) magnusde93@tg-dw222 ~/Documents/tof java hd1
-1794967296
3.14
0.0
3.14
0.0
```

```
1 a = 50000
2 b = 50000
3
4 print(a*b)
5
6 x = 3.14
7 y = 1e20
8 z = -1e20
9
10 print((x+y)+z)
11 print(x+(y+z))
```

(base) magnusde93@tg-dw222 ~/Documents/tof py hd1.py
2500000000
0.0
3.14

Ástæðan fyrir því að heiltölur eru mismunandi á Java og Python er að Java tekur bara in 32 bita heiltökur en Python er með arbitrary size integer sem þýðir að þú getur notað óákveðið langa heiltölu en hinsvegar hefur þetta áhrif á keyrslutíma.

3.

a. Stallarnir sýna hversu mikið af upplýsingum komast fyrir í hvert minnissvæði. L1 er hraðasta minnissvæðið og er því best að vera með allar upplýsingar í því minnissvæði. Ef upplýsingarnar eru meiri en komast fyrir í minnissvæði L1 þá er tekið upplýsingar úr L2 og hent einhverjum upplýsingum úr L1 sem samsvarar stærð upplýsinga sem kemur úr L2. CPU sækir upplýsingar í L2, L3, L4 eftir þörfum og hendir þá upplýsingum úr L1 til þess að gera pláss fyrir upplýsingar í L1 sem verið er að nota.

b.

500mb á móti 14000mb eða 1/28.