Assumption University
Faculty of Engineering
Semester 1/2018

CE3111 Design And Analysis of Algorithms
Term Project

1756 One and a Half Digger
(Timus Online Judge)

Submit to
Asst. Prof. Dr. Thitipong Tanprasert

by
Win Shaine
5825401

Nov 29, 2018
Problem:

Vitya Perestukin is solving the following problem: Three diggers can dig a trench in exactly one day. How many diggers are needed to dig the same trench in exactly two days? Vitya has concluded that one and a half diggers are needed. But there can't be such an answer. Actually, two diggers are needed: on the first day only one digger will work, and on the second day they both will work. It is known that m diggers can dig a trench in exactly d1 days if they all work every day. Help Vitya compile a work schedule according to which a minimal number of diggers can dig a trench in exactly d2 days. Difficulty: 162

Input
The only input line contains the integers m, d1, and d2 (1 ≤ m, d1, d2 ≤ 10 000).

Output
In the only line output d2 integers, which are the numbers of diggers that should work on each of the days so that the trench will be dug in time. It is possible that on some days (including the last day) nobody will work. If there are several solutions, output any of them.

Sample

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 1 2</td>
<td>1 2</td>
</tr>
</tbody>
</table>
Problem Solution:

```python
import sys
x = input().split()
m = int(x[0])
d1 = int(x[1])
d2 = int(x[2])

if m < 1:
    sys.exit()
if d1 & d2 >= 10000:
    sys.exit()

for_one_day = m*d1

mini_worker = for_one_day/d2
max_worker = (for_one_day+d2-1)/d2

max_worker_day_count = for_one_day/d2
print('max_worker_day',max_worker_day_count)
mini_worker_day_count = d2 - max_worker_day_count
print('mini_worker_day',mini_worker_day_count)

for i in range(max_worker_day_count):
    print('max_worker',int(max_worker))

for j in range(mini_worker_day_count):
    print('mini_worker',int(mini_worker))
```

Line 13: I calculated Maximum Work Power to finish the job
Line 15: Calculation Minimum worker need for a day for d2
Line 16: Calculation Maximum worker for a day for d2
Line 18: Maximum worker working day
Line 19: Minimum worker working day
Test Cases:

```python
$ python3 onehalfdiggers.py
3 1 2
max_worker_day 1
mini_worker_day 1
max_worker 2
mini_worker 1

$ python3 onehalfdiggers.py
3 3 6
max_worker_day 3
mini_worker_day 3
max_worker 2
max_worker 2
max_worker 2
mini_worker 1
mini_worker 1
mini_worker 1

$ python3 onehalfdiggers.py
5 5 6
max_worker_day 1
mini_worker_day 5
max_worker 5
mini_worker 4
mini_worker 4
mini_worker 4
mini_worker 4
mini_worker 4

$ python3 onehalfdiggers.py
3 2 9
max_worker_day 6
mini_worker_day 3
max_worker 1
max_worker 1
max_worker 1
max_worker 1
max_worker 1
mini_worker 0
mini_worker 0
mini_worker 0
```
Submission Result:

Problem: 1756
Language :Python 3.6
Execution time:0.109
Memory used: 608KB

<table>
<thead>
<tr>
<th>ID</th>
<th>Date</th>
<th>Author</th>
<th>Problem</th>
<th>Language</th>
<th>Judgement result</th>
<th>Test #</th>
<th>Execution time</th>
<th>Memory used</th>
</tr>
</thead>
<tbody>
<tr>
<td>8153611</td>
<td>21-14-03 20Nov 2018</td>
<td>winshaine</td>
<td>1756, One and a Half Diggers</td>
<td>Python 3.6</td>
<td>Accepted</td>
<td>0.109</td>
<td>608 KB</td>
<td></td>
</tr>
</tbody>
</table>