Shows how the configuration is created by createVM() for sum.c test driver
The loop shows each VM word being written to with a value, and how each (the first several anyway) iteration of the loop translates to what is accessed)

Page Size = 4 words (4x4=16 bytes)
VM = 64x4 = 256 words (256x32=1024 bytes)
PM = 4x4 = 16 words (16x4=64 bytes)
(TLB Size = 0 entries)

```
 for (i = 0; i < MEMSIZE; i++)
 {
   writeInt(h, i, i);
 }
```

Accessing word 0, 1, 2, 3, 4, 5, 6, ...

(words 0, 1, 2, 3)
VM Page 0, 0, 0, 0
PM Page 0, 0, 0, 0

(words 4, 5, 6, 7)
VM Page 1, 1, 1, 1
PM Page 1, 1, 1, 1

(words 8, 9, 10, 11)
VM Page 2, 2, 2, 2
PM Page 2, 2, 2, 2

(words 12, 13, 14, 15)
VM Page 3, 3, 3, 3
PM Page 3, 3, 3, 3

(words 16, 17, 18, 19)
Page Table 1, 2, 3, 0-> Page Fault
VM Page 4, 4, 4, 4
PM Page 0, 0, 0, 0

(words 20, 21, 22, 23)
Page Table 2, 3, 0, 1 --> Page Fault
VM Page 5, 5, 5, 5
PM Page 1, 1, 1, 1