Practice VM Question

- Our system has the following properties
  - 1 MiB of physical address space $m=20$
  - 4 GiB of virtual address space $n=32$
  - 32 KiB page size $p=15$
  - 4-entry $\underbrace{\text{fully associative}}_{1 \text{ set}}$ TLB with LRU replacement

a) Fill in the following blanks:

2^{17} Total entries in page table

2^{n-p} ← # of virtual pages

20 Minimum bit-width of PTBR ← physical address of PT

m

17 TLBT bits

VPN → TLBT / TLBI

here TLBI = 0

2^{5} Max # of valid entries in a page table ← # of pages in physical memory

2^{m-p}

Practice VM Question

starting address of matrix is at page offset of 0

- One process uses a page-aligned *square* matrix `mat[]` of 32-bit integers in the code shown below:

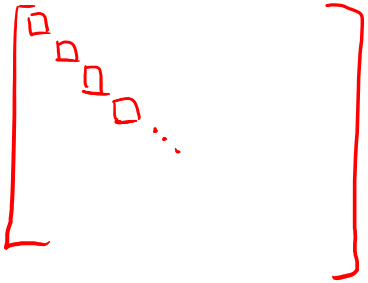
```
#define MAT_SIZE = 2048
for(int i=0; i<MAT_SIZE; i++)
      mat[i*(MAT_SIZE+1)] = i;
```

updating diagonal entries

b) What is the largest stride (in bytes) between successive memory accesses (in the VA space)?

stride is always 2049 ints = 2049*4 bytes

i	array index accessed
0	0
1	2049
2	2*2049
⋮	⋮



Practice VM Question

page size = 32 KiB = 2^{15} B

- One process uses a page-aligned *square* matrix `mat[]` of 32-bit integers in the code shown below:

```
#define MAT_SIZE = 2048
for(int i=0; i<MAT_SIZE; i++)
     mat[i*(MAT_SIZE+1)] = i;
```

2^{11} ints = 2^{13} B

c) What are the following hit rates for the *first* execution of the for loop? (assume all of mat[] starts on disk)

3/4 = 75% TLB Hit Rate

0% Page Table Hit Rate

access pattern: single write to index
never revisit indices (always increasing)
we access every row of matrix exactly once

each page holds $2^{15}/2^{13} = 4$ rows of matrix

within each page: M H H H

only access PT on TLB Miss

because mat[] on disk, each first access to page causes page fault.