You may work in pairs for this assignment. If you choose to work with a partner, make sure only one of you submits a solution, and that the file lists names and PIDs for both of you at the beginning.

Prepare your answers to the following questions in a plain text file. Submit your file to the Curator system by the posted deadline for this assignment. No late submissions will be accepted. For all questions, show supporting work if you want partial credit.

You will submit your answers to the Curator System (www.cs.vt.edu/curator) under the heading `MIPS04`.

For the following questions, we use the IEC definitions of the following terms: 1 kibibyte (KiB) is 2^{10} bytes, 1 mebibyte (MiB) is 2^{20} bytes, 1 gibibyte (GiB) is 2^{30} bytes and one tebibyte (TiB) is 2^{40} bytes.

1. Consider a two-dimensional, 128 x 128 array of values of type **`double`**. Suppose we have such three arrays, `A`, `B`, and `C`. Assume that each array is laid out in memory in row-major order, and the beginning of each array is aligned to a page boundary (e.g., `A[0][0]` is located at the beginning of a page; `B[0][0]` starts at another page; etc.). Consider the following matrix multiplication (ijk) algorithm:

```
for (int i = 0; i < 128; i++) {
    for (int j = 0; j < 128; j++) {
        double sum = 0.0;
        for (int k = 0; k < 128; k++) {
            sum += A[ i ][ k ] * B[ k ][ j ];
        C[ i ][ j ] = sum;
    }
}
```

Assume that the variable `sum` is register allocated (i.e., an access to `sum` does not lead to a memory access, and thus has no virtual-to-physical page translation);

Assume the virtual memory system uses 4 KiB pages, and that the number of virtual and physical pages is large enough that no page replacements must be performed.

a) [6 points] How many pages are needed to store each array?

b) [6 points] By the time the first sum for `C[0][0]` is calculated and stored, how many page faults will have been triggered 1) by accessing the first row of array `A`; 2) by accessing the first column of array `B`; and 3) by performing a write to `C[0][0]`?

c) [6 points] How many page faults will have been triggered by array element accesses by the time the multiplication finishes?

2. Consider a system that uses 48-bit virtual addresses.

a) [5 points] What is the maximum size, in bytes, of the virtual address space for a process, if the system uses 4KiB pages? Express your answer in TiB.

b) [5 points] What is the maximum size, in bytes, of the virtual address space for a process, if the system uses 2MiB pages? Express your answer in TiB.

c) [5 points] What is the maximum number of virtual pages a process might have, if the system uses 4KiB pages? Express your answers in terms of powers of 2.

d) [5 points] What is the maximum number of virtual pages a process might have, if the system uses 2MiB pages? Express your answers in terms of powers of 2.

3. Answer each of the following questions in one or two sentences.

a) [5 points] How do operating systems or hardware find the location of the page table of each process?

b) [5 points] Does a TLB miss always imply a page fault? Justify your answer.

c) [6 points] Explain one disadvantage of a physically-addressed cache in which physical addresses are used for cache set lookup and tag comparison.

d) [6 points] Explain one disadvantage of a virtually-addressed cache in which virtual addresses are used for cache set lookup and tag comparison.

4. Suppose a system uses 16-bit physical and virtual addresses, with a page size of 4 KiB. The system use a page table to maintain the mapping of virtual page numbers to physical page numbers. Suppose that, for a particular process, the current page table is:

Valid?	Dirty?	VPN	PPN/Disk	
1	1	0	6	Virtual page 0 is in DRAM in physical page 5
0	0	1	Disk	Virtual page 1 is on disk, not in DRAM
1	0	2	5	...
1	0	3	Disk	
1	0	4	7	
0	1	5	4	
...	...	...	...	More, irrelevant PTEs not shown

The system also uses a translation lookaside buffer (TLB) to cache recently-accessed page table entries (PTEs); the TLB uses pure LRU replacement; and the TLB can holds up to four PTE entries. Suppose that, for the same particular process, the current TLB is:

Valid?	Dirty?	VPN	PPN	Accessed	
1	1	0	6	t1	Virtual page 0 is in physical page 5, last access was time 1
1	1	5	4	t2	...
1	0	2	5	t3	
0	0	8	3	t0	This entry is invalid (stale from a previous process).

Assume that if a page must be brought from disk, the next available physical pages would be 8, 9, etc.

a) [4 points] Suppose that at time `t4` the process performs a write operation to virtual address `0x0123`. What would the corresponding physical address be?

b) [12 points] After the write to virtual address `0x0123` at time `t4`, show the contents of 1) the page table and 2) the TLB. Then, specify if this memory access results in 3) a TLB miss; and/or 4) a page fault.

c) [12 points] Suppose that at time `t5` the process performs a read operation from virtual address `0x4321`. Show the contents of 1) the page table and 2) the TLB. Then, specify if this memory access results in 3) a TLB miss and/or 4) a page fault.

e) [12 points] Suppose that at time `t6` the process performs a read operation from virtual address `0x3030`. Show the contents of 1) the page table and 2) the TLB. Then, specify if this memory access results in 3) a TLB miss. and/or 4) a page fault.