1. Assume the following memory system:
   - Virtual addresses are 14 bits
   - Physical addresses are 12 bits
   - The page size is 64 bytes
   - The first 16 entries in the page table are as follows: (VPN means virtual page number, PPN means physical page number)

VPN	PPN	Valid
00	28	1
01	-	0
02	33	1
03	02	1
04	-	0
05	16	1
06	-	0
07	-	0

	PPN	Valid
08	13	1
09	17	1
0A	09	1
0B	-	0
0C	1E	1
0D	2D	1
0E	11	1
0F	0D	1

(1) What is the format of a virtual address? In other words, how many bits are used for the virtual page number and how many bits are used for the virtual page offset?

(2) What is the format of a physical address? In other words, how many bits are used for the physical page number and how many bits are used for the physical page offset within a page?

(3) Show how this memory system translates the virtual address 0x030D into the corresponding physical address. **Write the tranlation process!**

Solution:

(1) Virtual page offset: 6 bits, virtual page number: 8 bits

(2) Physical page offset: 6 bits, physical page number: 6 bits

(3) 0x030D= $(00\ 0011\ 0000\ 1101)_2$, so the virtual page number is $(00001100)_2$=0x0C. Use this VPN to lookup the page table, get the PPN=1E.

So the physical address = $(011110001101)_2$=0x78D

2. Consider the memory system with the following specifications:
   - Byte-addressable
   - Virtual address space: 4G bytes
   - Main memory size: 16M bytes
   - Cache size: 256K bytes
   - Page size: 64K bytes
   - Block size: 128bytes
   - Mapping Strategies: Main Memory to Cache: 4-way set associative; Hard Disk to Main Memory: fully associative

   Virtual address is first translated to physical address. Then, it accesses the cache memory using the physical address.

   (1) How many sets are there in the cache memory?

   (2) How long is the tag field of the cache?

   (3) Given a virtual address 0B45DA12 (hexadecimal), its corresponding virtual page is stored in physical page 3E (hexadecimal).

   ① What is its physical address under such mapping?

   ② Which set can this address be possibly found in the cache?

   ③ Which byte does this address point to out of the 128 bytes in a block?

Solution:

(1) (256K/128)/4=512

There are 512 sets in the cache memory.

(2) Because the main memory is byte-addressable and the main memory size: 16M bytes, so the main memory address is 24-bit long. From (1) there are 512 sets in the cache memory, so the set field of the main memory address is 9-bit long. And because the block size: 128 bytes, so the byte field of the main memory address is 7-bit long. Finally, the tag field of the cache is 24－9－7 =8-bit long.

(3) ①There are 4G/64K=2^{16} virtual pages ,so the virtual page number is 16-bit long. Because the virtual address is 32-bit long, so the offset field of it is 32-16=16-bit long.

The virtual address 0B45DA12 can be divided into two parts, the highest 16-bit virtual page number and the lowest 16-bit offset. So the offset field of this virtual address is DA12. Concatenate it with the physical page number. So the physical address under such mapping is 3EDA12.

②3EDA12=$(0011\ 1110\ \underline{1101\ 1010}\ 0001\ 0010)_2$

So this address can be found in $1B4_{16}$ or 436 set in the cache.

③3EDA12=$(0011\ 1110\ 1101\ 1010\ 0\underline{001\ 0010})_2$

The byte number is 18.

3. Consider the two-dimensional array A:

int A[][] = new int[100][100];

Where A[0][0] is at location 200, in a paged memory system with pages of size 200. A small process is in page 0 (locations 0 to 199) for manipulating the matrix; thus, every instruction fetch will be from page 0.

Suppose that the elements of the array are stored in row order in the virtual memory. For three page frames, how many page faults are generated by the following array-initialization loops, using LRU replacement, and assuming page frame 1 has the process in it, and the other two are initially empty?

(1)
```
for (int j = 0; j < 100; j++)
    for (int i = 0; i < 100; i++)
        A[i][j] = 0;
```

(2)
```
for (int i = 0; i < 100; i++)
    for (int j = 0; j < 100; j++)
        A[i][j] = 0;
```

Solution:

(1) 5000 (2) 50