1. (5 points) Circle the correct answer among the choices given. If you circle more than one answer, you will lose the grade of the corresponding question.

(A) The range of numbers that can be present by signed int and unsigned int are the same.
   a. True  b. False  c. depends operating systems

(B) We can have two caches of the same size but different associativity
   a. The above statement is true.
   b. The above statement is false.
   c. It depends on whether we have TLB or not.

(C) What is the correct order for memory access
   a. TLB-page table- cache  b. cache-page table -TLB
   c. page table- cache - TLB  d. page table - TLB - cache

(D) The dynamic memory allocator deals with virtual addresses only:
   a. True  b. False  e. Depends on the size of the cache

(E) Assume the register rax stores value x. What will be the content of rax after executing the instruction: \texttt{leaq 0(\%rax, \%rax, 2), \%rax}
   a. 3x  b. 4x  c. 5x  d. depends on what is stored in memory
2. [12 points] Given the following x86_64 assembly code and its corresponding C code, fill-in the blanks in the corresponding C code. Also on the far right, fill in the correspondence between each register and its corresponding variable in C. (Hint: you can neglect edx because it does not correspond to any variable in the C code and is used only by the compiler for the translation).

| do: movl $2, %eax | int do(int x, int y){
| movl $32, %ebx |   int j =________;
| xorl %ecx, %ecx |   int data =_______;
| L0: cmpl %ebx, %ecx |   int i;
|   jg L1 |   i = __________;
|   cmpl %edi, %eax |   while(______________){
|     jle L2 |     if( data <= x){
|     addl %esi, %eax |       ____________;
|     jmp L3 |     else
| |       ____________;
| |     }
| |   ____________;
| L2: movl %edi, %edx |   return data;
|     addl %esi, %edx | }
|     addl %edx, %eax | }
| L3: addl $2, %ecx | int do(int x, int y){
|     jmp L0 |   int j =________;
| L1: Ret |   int data =_______;
| |   int i;
| |   i = __________;
| |   while(______________){
| |     if( data <= x){
| |       ____________;
| |     else
| |       ____________;
| |     }
| |   ____________;
| | return data;
| | }

3. [1 point] From the assembly code above, in one sentence explain why we used the e version of registers (e.g. eax, ebx, ...) instead of the q version (like rax, rbx, ...)?

4. [2 points] Write two different instructions to multiply register rax by 4 without using multiplication instructions. Each one must be one assembly instruction only. Assume rax contains unsigned long

- 
- 

- 

-
5. For the following piece of code:

```c
1. void par() {
2.     if (fork() == 0) {
3.         fork();
4.         printf("hello\n");
5.     }
6.     return;
7. }
8.
9. int main() {
10.    printf("hello\n");
11.    par();
12.    printf("hello\n");
13.    exit(0);
14. }
```

a. [1 point] How many times will “Hello” be printed?

b. [1 point] Before executing line 13 by any process; how many page tables exist (assuming the system uses 1-level page table)?

c. [3 points] If instead of `return` at the end of `par()` function we use `exit(0)`. Will your answer on question (a) above differ? If yes, what is the new answer? If no, explain why not.
6. Suppose that we have a system with main memory access time of 100 cycles. We added to that system a cache with 50 cycles access time. The resulting average memory access time becomes 150.

a. [1 points] Did we benefit from having a cache in that case? Why?

b. [1 points] What is the cache miss rate?

c. [1 point] Is the cache mentioned above accessed with virtual address? or physical address?

d. [2 points] Suppose the above cache has 6 bits for offset, 4 bits for set, and 22 bits for TAG. What is the total cache size in bytes assuming the cache is 2-way set associative? You can leave the size as power of two to make it easier. To get full-credit, show all the steps.