1. (10) Consider the assembly code for the function Insert() on p.13 of the PLN unit on the JVM. The instruction aload_0 in offset 78 will push ______ onto the stack. (Your answer must be a variable name or a Java reserved word.)

2. (10) Consider these Java functions:

```java
class Example {
    public int sum2(int a, int b) {
        return a + b;
    }
    public int sum3(int u, int b, int c) {
        return a + b + c;
    }
    public int sum4(int u, int b, int c, int d) {
        return a + b + c + d;
    }
}
```

The compiled code for `sum3()` is

```assembly
0: iload_1
1: iload_2
2: iadd
3: iload_3
4: iadd
5: ireturn
```

The size of the compiled code for `sum2()` has ______ more bytes than that of `sum3()`, while the size for `sum4()` has ______ more bytes than that of `sum3()`. (Your answers must be positive, negative or zero integers.)

3. (15) The three lines in Tetris.s beginning at line ______ enable you to continue with other work after finishing your game.

4. (15) Suppose x.s has a label zzz in the .text segment, and we wish execution to start there. We assemble the file, producing x.o. Show the ld command we would need so as to arrange execution to start at zzz.

5. (20) A certain Java function is static and has no arguments, but has three local variables, declared as

```java
int x, y, z;
```

Fill in the following blanks in the code below, which does

```java
z = (x+y) * (x+y+2);
```

```assembly
iload_0
iload_1
---------
iconst_2
iadd
---------
---------
```

6. (10) A disadvantage of a virtually-mapped cache is that we could not have blocks from different ______ simultaneously.

7. (20) The following code prints out a 2-hex-digit number in hex. Fill in the blanks:

```bash
# prints out a 2-hex-digit number in hex
printx:
    # 3 pushl, not shown
    movl 16(%esp), %eax
```
movl $0, %edx
movl $0x10, %ecx
idivl %ecx  # quotient, remainder in EAX, EDX
pushl %eax
call printhexdig
pushl ________
call printhexdig
addl $8, %esp
# 3 popl, not shown
ret

printhexdig:
# 5 pushl, not shown
movl 24(%esp), %esi
cmpl ________, %esi
jge caseaf
addl $'0', %esi
jmp prnt
caseaf:
subl $10, %esi
addl $'a', %esi
prnt:
    movl $4, %eax
    movl $1, %ebx
    pushl %esi
    movl ________, %ecx
    movl $1, %edx
    int $0x80
# 6 popl, not shown
ret

Solutions:

1. this

2. -2, 3; for the latter, note that we’ll need an iload instruction (with operand 4), since there is no such thing as iload_4

3. 632 (three lines after gameover beginning with getterm)

4. ld x.o -e zzz

5. iadd, dup, imul, istore_2

6. processes

7. %edx, $10, %esp