1. a)
```
li      $t0, 5
mul     $t0, $t0, 4    # since ints are 4 bytes
lw      $t0, arr($t0)
move    $a0, $t0
li      $v0, 1
syscall
```

   b)
```
li      $t0, 5
mul     $t0, $t0, 4    # since ints are 4 bytes
lw      $t1, arr($t0)
sw      $t1, k
add     $t1, $t1, 1
sw      $t1, arr($t0)
```

   Note that a smart compiler would have probably multiplied the 5 by the 4 during the optimization phase, and replaced the first two instructions by `li $t1, 20`, but we're assuming that no optimization is performed.

   c)
```
lw      $t0, k
add     $t0, $t0, 5
mul     $t0, $t0, 3
mul     $t0, $t0, 4
lw      $t0, arr($t0)
sw      $t0, k
```

2. a)
```
li      $t0, 2008
li      $t1, 8
sw      $t0, d($t1)
```

   b)
```
li      $t0, 0
lw      $t0, d($t0)
li      $t1, 4
lw      $t1, d($t1)
add     $t0, $t0, $t1
move    $a0, $t0
li      $v0, 1
syscall
```

   c)
```
# this is the array subscript
li      $t0, 15
# multiply the subscript by the
# size of Date structure (12 bytes)
mul     $t0, $t0, 12
# add the offset of the year field
add     $t0, $t0, 8
lw      $t0, may($t0)
sw      $t0, n
```

3. a)
```
la      $t0, x
sw      $t0, p
```

b)
```
li      $t0, 13
lw      $t1, p
sw      $t0, ($t1)
```

c)
```
lw      $t0, p
lw      $t0, ($t0)
move    $a0, $t0
li      $v0, 1
syscall
```

d)
```
la      $t0, p
sw      $t0, q
```

e)
```
lw      $t0, q
lw      $t0, ($t0)
lw      $t0, ($t0)
sw      $t0, y
```

f)
```
        lw      $t0, p
        beqz    $t0, done
        li      $t1, 212
        sw      $t1, ($t0)
done:
```

where "`done`" is the label of whatever instruction follows the `if` statement.

g)
```
        lw      $t0, q
        beqz    $t0, done
        lw      $t0, ($t0)
        beqz    $t0, done
        li      $t1, 212
        sw      $t1, ($t0)
done:
```

where "`done`" is the label of whatever instruction follows the `if` statement.

4.
```
add     $sp, $sp, 4
lw      $ra, ($sp)
jr      $ra
```

5. a)

```
        .text
f:      sw      $ra, ($sp)
        sub     $sp, $sp, 4
        li      $a0, 1
        li      $v0, 1
        syscall
        add     $sp, $sp, 4
        lw      $ra, ($sp)
        jr      $ra

g:      sw      $ra, ($sp)
        sub     $sp, $sp, 4
        jal     f
        add     $sp, $sp, 4
        lw      $ra, ($sp)
        jr      $ra

main:   li      $sp, 0x7ffffffc
        jal     f
        jal     g
        li      $v0, 10
        syscall
```

b)

```
        .text
f:      sw      $ra, ($sp)
        sub     $sp, $sp, 8
        li      $t0, 5
        sw      $t0, 4($sp)
        move    $a0, $t0
        li      $v0, 1
        syscall
        add     $sp, $sp, 8
        lw      $ra, ($sp)
        jr      $ra

main:   li      $sp, 0x7ffffff8
        li      $t0, 3
        sw      $t0, 4($sp)
        move    $a0, $t0
        li      $v0, 1
        syscall
        jal     f
        li      $v0, 10
        syscall
```

6.

```
#include <stdio.h>

void f(int x) {
  int y;
  if (x != 0) {
    printf("%d", x);
    y= x - 2;
    f(y);
  }
}

int main() {
  f(6);
  return 0;
}
```