9.1
a) [5, 2, 4]
b) [4, 5, 5]
c) [4, 5, 5]

9.3
I'll answer these questions using the following non-tail recursive fibonacci function in OCaml:

```ocaml
# let rec fib n =
if n < 3 then 1
else fib(n-1) + fib(n-2);;
```

da) In the topmost activation record at the time of the function call, the parameter \( n \) is the value of \( n \) in the first call to the function. So if the call is \( \text{fib}(9) \), the parameter \( n \) in the topmost activation record is 9. Lower activation records will have values of \( n \) upon which the first function call depends.

db) The function to calculate the number of activation records required to determine \( \text{fib}(n) \) is the fibonacci function! There are 213 activation records required to calculate \( \text{fib}(13) \).

dc) The OCaml translation of this function is:

```ocaml
# let rec gcd (x, y) =
if (y=0) then x
else gcd(y, x mod y);;
```

The activation records would look like:

```
| x=24, y=10 |
| x=24, y=10 |
| x=24, y=10 |
| x=10, y=4  |
| x=10, y=4  |
| x=4,  y=2  |
```

return 2