1. (20) The code below subtracts 1 from EAX if the contents of that register are negative, but leaves EAX unchanged otherwise. Fill in the blanks:

```
subl $0,%eax
js done
decl %eax
done: ...
```

2. This problem concerns the sorting example on pp. 72ff.

(a) (20) Give the line number of the instruction executed after the one in line 88.

(b) (20) The first time 35 is executed, what specific value will be in EDI?

Parts (c) and (d) involve the following context:

Suppose in executing the program from within GDB, I set a breakpoint at line 76 (a js instruction), and run the program. When the program reaches line 76, I issue the command `info registers`, and get the following output:

```
eax      0x80490c4
ecx      0x1
edx      0x80490d0
ebx      0x6
ebp      0x4
esi      0x80490cc
edi      0x2
eflags   0x287
```

The columns here give name of the register, and its contents in hex. (I've omitted some material.)

(c) (20) Let’s refer to the loop in lines 31-42 as the “main” loop. What iteration of that loop was the program in when the current call to `findmin()` was executed? Answer first, second etc.

(d) (20) Based purely on the above information above and NOT on the specific values in the array x (1,5,2,...), state whether the jump will be taken or not, i.e. whether we’ll jump to line 82. Explain which specific quantity (not just which register) shows this.

Solutions:

1.

```
subl $0,%eax
js done
done: ...
```