Accumulator Machine Exercise 1

For the following Accumulator machine program, 1) build the symbol table (pass1) and 2) translate the assembly code to machine code (pass 2)

comment(` first example accumulator machine program `)
comment(` `)
comment(` data section for program -- word(label,value) `)
word(a, 23)
word(b, 45)
word(c, 17)
word(d, 0)
comment(` code that implements the expression d = a + b - c; `)
label(start)
load(a) comment(` ACC <- memory[a] `)
add(b) comment(` ACC <- ACC + memory[b] `)
sub(c) comment(` ACC <- ACC - memory[c] `)
store(d) comment(` memory[d] <- ACC `)
halt
comment(` start execution at label start `)
end(start)
Accumulator Machine Exercise 2

Give the symbol table and the translation for the following accumulator machine program:

```plaintext
comment(" another example accumulator machine program ")
comment(")
comment(" computes answer = x + 5 * y ")
comment(" partial data set for program -- word(label, value) ")
word(x, 7)
word(five, 5)

comment(" code that implements the expression answer = x + 5 * y; ")
label(begin)
load(y) comment(" ACC <- memory[y] ")
mul(five) comment(" ACC <- ACC * memory[five] ")
add(x) comment(" ACC <- ACC + memory[x] ")
store(result) comment(" memory[result] <- ACC ")

halt

comment(" remaining data section")
word(y, 2)
word(result, 0)

comment(" start execution at label begin ")
end(begin)
```