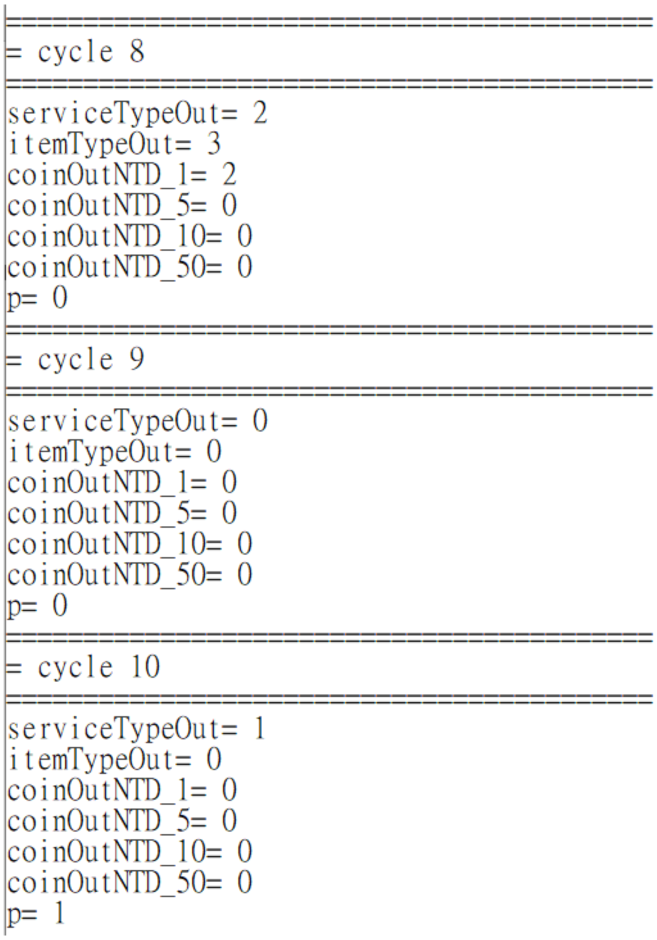
In the input: input-cex.pattern, we have only one pattern, which require 3 dollars for the charge. However, as we may see in the output, the vending machine does not have enough charge, and end up to "eat money".



```
==========================================
= cycle 8
==========================================
serviceTypeOut= 2
itemTypeOut= 3
coinOutNTD_1= 2
coinOutNTD_5= 0
coinOutNTD_10= 0
coinOutNTD_50= 0
p= 0
==========================================
= cycle 9
==========================================
serviceTypeOut= 0
itemTypeOut= 0
coinOutNTD_1= 0
coinOutNTD_5= 0
coinOutNTD_10= 0
coinOutNTD_50= 0
p= 0
==========================================
= cycle 10
==========================================
serviceTypeOut= 1
itemTypeOut= 0
coinOutNTD_1= 0
coinOutNTD_5= 0
coinOutNTD_10= 0
coinOutNTD_50= 0
p= 1
```

Fig 1. The output of simulation, where the assertion p = 1

After checking the source code of vending-simple.v, we find that the code in line 236:

serviceTypeOut_w     = `SERVICE_OFF;

should be to satisfy the spec of the origin design:

serviceTypeOut_w     = `SERVICE_BUSY;

By this small change, we can fix the bug of shortage of change.