```
def generate_literal_table(assembly_code):

    ans=[]
    literal_table = {}

    # First Pass: Collect literals and their values
    current_address = None
    for line in assembly_code:
        parts = line.split()
        if parts[0] == "START":
            current_address = int(parts[1])
        elif parts[0] != "END" and parts[0]!='LTORG':
            for part in parts:
                if part.startswith("='"):
                    literal = part.strip("='")
                    literal_table[literal] = None
            current_address += 1
        elif parts[0]=='LTORG':
           for literal in literal_table:
              literal_table[literal] = current_address
              current_address += 1
           ans.append(literal_table)
           literal_table={}

    for literal in literal_table:
        if(literal_table[literal] == None):
          literal_table[literal] = current_address
          current_address += 1
    ans.append(literal_table)
    literal_table={}


    # Second Pass: Generate Literal Table
    print("Literal Table:")
    print("Literal\t|\tValue")
    print("----------------------")
    for i in range(len(ans)):
      for literal, value in ans[i].items():
          print(f"{literal}\t|\t{value}")

# Assembly code
assembly_code = [
    "START 200",
    "READ X",
    "READ Y",
    "MOVER AREG, ='5'",
    "MOVER BREG, ='6'",
    "ADD AREG, BREG",
    "LOOP MOVER CREG, X",
    "ADD CREG, ='1'",
    "COMP CREG, Y",
    "BC LT, LOOP",
    "LTORG",
    "NEXT SUB AREG, ='1'",
    "COMP AREG, Y",
    "BC GT, NEXT",
    "STOP",
    "X DS 1",
    "Y DS 1",
    "END"
]

# Generate literal table
generate_literal_table(assembly_code)
```

```
Literal Table:
Literal |       Value
----------------------
5       |       209
6       |       210
1       |       211
1       |       218
```