Data: A set $C = \{c_1, c_2, \ldots, c_r\}$ of denominations of coins, where $c_i > c_2 > \ldots > c_r$ and a positive number n
Result: A list of coins $d_1, d_2, \ldots, d_k$, such that $\sum_{i=1}^{k} d_i = n$ and k is minimized

```
C ← ∅;
[?] plain)
for i ← 1 to r do
    while n ≥ c_i do
        C ← C ∩ {c_i};
        n ← n − c_i;
    end
end
return C;
```

Algorithm 1: CHANGE Makes change using the smallest number of coins

```
Data: A sequence of integers (a_1, a_2, ..., a_n)
Result: The index of first location with the same value as in a
        previous location in the sequence
location ← 0;
i ← 2;
while i ≤ n and location = 0 do
    ;   /* Do the following if i is less than or equal to n */
    j ← 1;
    while j < i and location = 0 do
        if a_i = a_j then
            location ← i;
        end
        else
            j ← j + 1;
        end
    end
    i ← i + 1;
end
return location;
```

Algorithm 2: FINDDUPLICATE