Iterative solution:

```
minPayment( X, c[n])                          /* Assume Cn given as an array */
[redacted]                                    each index 0≤i≤n
2. initialize A[n+1]                          reeresent an amount of
3. do A[0] ← 0,                               Payment */
4. for amount ← 1 to X:
    5 do min ← X+1
    6. for coinIndex ← 0 to n do

            j ← amount − C[coinIndex]

            if 0 <= j and A[ j ] < min then do

                min ← A[j]

    7. do A[amount] ← 1+ min
                         same meaning as in f

8. if A[X] = X+1 then print "we can't pay an amount
       of X with any number of coins given the
       coins in c[n]"     /* should return NIL, but I
9. else return A[amount]      chose to write this, to make
                              the meaning of "A[X]= X+1"
                              clearer */
```

Notice that after we have calculated A[i] for some i then we don't need to calculate A[ℓ] again, for any $\ell \leq i$

Time complexity = $O(n \cdot X)$

Space complexity = $O(X)$