Given a sum s(amount) = 1 040 508 NOK and 4 Norwegian coins: {1,5,10,20}, we will write a dynamic programming algorithm using tabular approach.

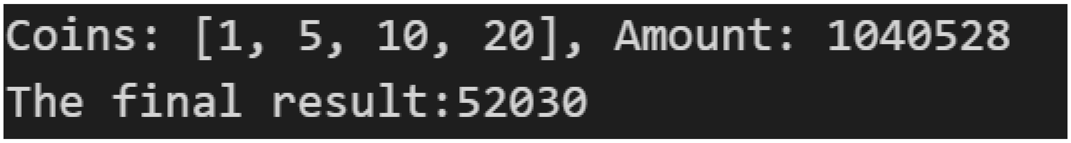
Tabular approach a technique is used for solving problems bottom-up, by filling out a table that has the length of the amount to store the results of subproblems. We will be solving smaller subproblems first and using their results to solve bigger problems, we will also avoid recalculating the same problem multiple times.

We start by creating a list called ‘table’ with a size of ‘amount + 1’. Each entry in this list is initialized to infinity (‘float('inf')’) to represent that we have not yet determined the minimum number of coins needed for each amount. We set ‘table[0] = 0’ because zero coins are required to make an amount of zero.

Next, we iterate over each amount ‘i’ from 1 up to the target ‘amount’. For each ‘i’, we examine all available coin denominations. If a coin does not cause the amount to become negative (i.e., ‘i - coin >= 0’), we check whether using this coin results in a solution that requires fewer coins than the current recorded minimum for ‘i’ in ‘table[i]’. This is done by comparing ‘table[i]’ with ‘table[i - coin] + 1’, where ‘table[i - coin] + 1’ represents the minimum number of coins needed for the smaller amount ‘i – coin’, plus the one coin we are currently considering.

After completing all iterations, ‘table[amount]’ will contain the minimum number of coins needed to make up the target amount. If ‘table[amount]’ is still infinity, it indicates that no combination of the given coins can produce the target amount, so we return ‘-1’.

Output of the minimum number of coins that sum up to s is:



```
Coins: [1, 5, 10, 20], Amount: 1040528
The final result:52030
```