COIN CHANGE

Question :

You are given an integer array coins representing coins of different denominations and an integer amount representing a total amount of money.

Return the fewest number of coins that you need to make up that amount. If that amount of money cannot be made up by any combinations of the coins, return -1.

You may assume that you have an infinite number of each kind of coin.

Ex 1:
Input : coins = [1, 2, 5], amount = 11
Output : 3
Explanation : 11 = 5+5 + 1

CODE

```
CONST INT N = 1e5+10;
VECTOR<INT> COINS (N);
INT DP[N];

INT FUNC (INT AMOUNT)
{
    IF (AMOUNT == 0) RETURN 0;
    IF (DP[AMOUNT] != -1) RETURN DP[AMOUNT];

    INT ANS = INT_MAX;
    FOR (INT COIN : COINS)
    {
        IF (AMOUNT - COIN >= 0)
        {
            ANS = MIN(ANS + 0LL, FUNC(AMOUNT
                                     - COIN) + 1LL);
        }
    }
    RETURN DP[AMOUNT] = ANS;
}

INT MAIN ()
{
    COINS = {1, 2, 5};
    INT AMOUNT = 11;
    MEMSET(DP, -1, SIZEOF(DP));
    INT ANS = FUNC(AMOUNT);
    COUT << (ANS == INT_MAX ? -1 : ANS)
                              << "\n";
    RETURN 0;
}
```

OUTPUT :

3