Theorem: (Linearity of Expectation) Let $R_0, R_1, \ldots$ be random variables s.t.

$$\sum_{\mathrm{i\,=\,0}}^{\infty} \mathbb{E}(|R_i|) \text{ converges}$$

then

$$\mathbb{E}(\sum_{\mathrm{i\,=\,0}}^{\infty} R_i) = \sum_{\mathrm{i\,=\,0}}^{\infty} \mathbb{E}(R_i)$$

In light of this theorem, an example where $\mathbb{E}(\sum_{\mathrm{t\,=\,0}}^{\infty} C_t) \neq \sum_{\mathrm{t\,=\,0}}^{\infty} \mathbb{E}(C_t)$ requires $\sum_{\mathrm{i\,=\,0}}^{\infty} \mathbb{E}(|R_i|)$ to diverge.

This question illustrates the Gambler's paradox which will be described next. Gambler's Paradox: Consider of a fair Roulette wheel. The expectation of the sum of your expected wins for each bet is zero ($\mathbb{E}(R_i) = 0$) Therefore,

$$\sum_{\mathrm{i\,=\,0}}^{\infty} \mathbb{E}(R_i) = 0$$

However, looking at the problem a different way, the probability of winning at least once as the number of spins approaches infinity is obviously one because it is a fair roulette wheel. Therefore, if a strategy can be devised that gaurentees that if the wheel lands on green and the player wins that the player can end the game and walk away with a net gain in money, then $\mathbb{E}(\sum_{\mathrm{i\,=\,0}}^{\infty} R_i) > 0$.

A well known strategy to achieve this is called "bet doubling." This strategy involves betting a starting amount, say 10, and every time the player loses they double their bet. If the player wins they stop playing and go home. The nth bet is $10 * 2^{n-1}$ whil the probability you make the bet (you haven't won yet) is 2^{-n} so $C_n = 10 * 2^{n-1}$ Therefore,

$$\mathbb{E}(|C_n|) = 10 * 2^{n-1} * 2^{-n} = 20$$
$$\implies \sum_{\mathrm{i\,=\,0}}^{\infty} \mathbb{E}(|C_i|) = 20 + 20 + \ldots$$

This means that $\sum_{\mathrm{i\,=\,0}}^{\infty} \mathbb{E}(|C_i|)$ diverges so the theorem does not hold so

$$\mathbb{E}(\sum_{\mathrm{t\,=\,0}}^{\infty} C_t) \neq \sum_{\mathrm{t\,=\,0}}^{\infty} \mathbb{E}(C_t)$$

In solving this I referenced https://eng.libretexts.org/Bookshelves/Computer$_Science/Programming_and_Computation_Fun$