Algorithm: $\boldsymbol{P} \leftarrow \texttt{CodeLlama}(\boldsymbol{x}|\boldsymbol{\theta})$

Input: $\boldsymbol{x} \in V^*$, a sequence of token IDs.
Output: $\boldsymbol{P} \in [0,1]^{N_\mathrm{V} \times \mathrm{length}(\boldsymbol{x})}$, where $\boldsymbol{P_t}$ represents the conditional distribution $p(x_{t+1} | \, \boldsymbol{x}_{1:t})$.
Hyperparameters: N_V, θ_max, L, H, d_e, d_mlp, MLP hidden dimension.
Parameters: $\boldsymbol{\theta}$ includes all the following parameters:

$\boldsymbol{W_e} \in \mathbb{R}^{d_\mathrm{e} \times N_\mathrm{V}}$	token embeddings matrix.
$\boldsymbol{W_p} \in \mathbb{R}^{d_\mathrm{e} \times \theta_\mathrm{max}}$	positional embeddings matrix.
$\boldsymbol{W}_l, \boldsymbol{W}_{e2d_l}$	attention and MLP weights.
For $l \in [L]$	See paper for details.
$\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^{d_\mathrm{e}}$	output normalization parameters.
$\boldsymbol{W_u} \in \mathbb{R}^{N_\mathrm{V} \times d_e}$	unembedding matrix.

$\theta \leftarrow \mathrm{length}(\boldsymbol{x})$
for $t \in \theta$: $\boldsymbol{e}_t \leftarrow \boldsymbol{W}_e[:, x[t]] \; + \; \boldsymbol{W}_p[:, t]$
$\boldsymbol{X} \leftarrow [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots \boldsymbol{e}_\theta]$
for $l = 1,2,\ldots,L$ **do**
    $\boldsymbol{X} \leftarrow \texttt{layer\_norm}(\boldsymbol{X}[:, t] \mid \boldsymbol{\gamma}_l^1, \boldsymbol{\beta}_l^1)$
    $\boldsymbol{X} \leftarrow \boldsymbol{X} + \texttt{MHAttention}(\tilde{X} \mid \boldsymbol{W_l}, \mathrm{Mask}[t, t'] = [[t \leq t']]$
    $\boldsymbol{X} \leftarrow \texttt{layer\_norm}(\boldsymbol{X} \mid \boldsymbol{\gamma}, \boldsymbol{\beta})$
    $\boldsymbol{X} \leftarrow \boldsymbol{X} + \mathrm{MLP}(\boldsymbol{X} \mid \boldsymbol{W_l})$
$\boldsymbol{X} \leftarrow \texttt{layer\_norm}(\boldsymbol{X} \mid \boldsymbol{\gamma}, \boldsymbol{\beta})$
end
for $t \in [\theta]$: $\boldsymbol{P}_t \leftarrow \mathrm{softmax}(\boldsymbol{W_u X}[:, t])$
return $\boldsymbol{P}$

This implements an autoregressive decoder-only transformer similar to GPT with causal self-attention masking. It processes the input sequence, applies multiple transformer layers with causal self-attention and MLP blocks, and produces a conditional distribution over the next token at each position. The specifics like layer normalization and multi-head attention are as defined in the "Formal Algorithms for Transformers" article.