Flow of a Large Language Model (LLM)

1. Input Processing

- **Text Input:** The user provides input in the form of a string (e.g., a question or instruction).
- **Tokenization:** The input is split into smaller components called tokens (words, subwords, or characters).
- **Token ID Mapping:** Each token is mapped to a unique ID from the model's vocabulary for further processing.

2. Encoding with Transformer Layers

- **Embedding Layer:** Tokens are converted into dense vectors representing semantic meaning.
- **Positional Encoding:** Since transformers don't have a built-in sense of word order, positional information is added to each token vector.
- **Self-Attention Mechanism:** The model evaluates relationships between tokens to identify which ones are most contextually relevant.
- **Feedforward Layers:** Vectors are passed through dense layers that transform the information.
- **Stacked Transformer Blocks:** Multiple layers (e.g., dozens in large models like GPT-3 or GPT-4) refine the representation and understanding of the input.

3. Output Generation

- **Logits Calculation:** After encoding, the model outputs a set of raw scores (logits) for each possible token.
- **Probability Distribution:** Logits are passed through a softmax function to create a probability distribution over the vocabulary.
- **Token Prediction:** The next token is selected based on this distribution.
- **Iterative Generation:** This process repeats token by token until a stopping condition is met (e.g., stop token, max token limit, or end-of-sequence signal).

4. Decoding Strategies

Used to control how the next token is selected:

- **Greedy Decoding:** Selects the most likely token at each step.
- **Beam Search:** Maintains multiple hypotheses to find the best output sequence.
- **Top-k Sampling:** Samples from the top-k most likely tokens.
- **Top-p (Nucleus) Sampling:** Samples from the smallest set of tokens whose cumulative probability exceeds a threshold (p).
- **Temperature Scaling:** Adjusts randomness — higher values → more creativity, lower values → more deterministic output.

---

5. Postprocessing

- **Detokenization:** Converts the sequence of tokens back into human-readable text.
- **Final Output Formatting:** Any special formatting, alignment, or structure is applied as per task requirements (e.g., translation, Q&A, summarization).

---

6. Optional Enhancements

- **Fine-Tuning / Instruction Tuning:** Additional training on specific tasks or instruction-following data to adapt the base model.
- **Reinforcement Learning from Human Feedback (RLHF):** Improves model behavior in terms of safety, helpfulness, and ethical alignment by incorporating human preferences.