**1. Instruções assembly utilizadas**

* **MOV**: Utilizada para movimentação de dados. Exemplo: mov DWORD PTR [ebp-16], 100. (Página 1 ​)
* **AND**: Realiza uma operação lógica AND bit a bit. Exemplo: and eax, 1. (Página 15​).
* **TEST**: Compara dois operandos logicamente usando AND, sem alterar o destino. Exemplo: test eax, eax. (Página 20​).
* **JE**: Instrução de desvio condicional (salto se igual). Exemplo: je .L3. (Página 22​).
* **ADD**: Realiza uma soma. Exemplo: add DWORD PTR [ebp-4], eax. (Página 7​).
* **JMP**: Salto incondicional. Exemplo: jmp .L2. (Página 21​).

**2. Endereços de memória associados às variáveis**

Os endereços de memória são representados como deslocamentos em relação ao ponteiro de base (ebp):

* [ebp-16]: Variável armazenando o valor inicial 100.
* [ebp-20]: Variável armazenando o valor 1.
* [ebp-4]: Usado como acumulador para a soma.
* [ebp-8]: Armazena um valor copiado de [ebp-20].
* [ebp-12]: Contador incrementado dentro do laço.

**3. Registradores utilizados**

* **EAX**: Registrador geral utilizado para manipulação e operações lógicas/aritméticas.
* **EBP**: Ponteiro de base para acessar variáveis locais.

**4. Constantes usadas**

* **100**: Carregada na instrução mov DWORD PTR [ebp-16], 100.
* **1**: Carregada em [ebp-20] e usada na operação lógica and eax, 1.
* **0**: Utilizada na inicialização de [ebp-4] e [ebp-12].