

## RESPOSTAS

Observe atentamente cada passo do processo e descreva-o com o máximo de detalhes que puder a execução do aplicativo 'Little Man Computing'.

**R:**

**1° CICLO** - *Program counter* envia sinal, passando pela *instruction register*, *address register* e chegando no *accumulator* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **1**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **0** a instrução **514**, no registro de instrução obtém-se o resultado **5** e no registro de endereço **14**, retornando até a casa **14** da memória obtendo **32**, seguindo pelo barramento este chega até o *accumulator* tendo como resultado **032**.

**2° CICLO** - *Program counter* envia sinal, passando pela *instruction register*, *address register* e chegando no *accumulator* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **2**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **1** a instrução **317**, no registro de instrução obtém-se o resultado **3** e no registro de endereço **17**, retornando até a casa **17** da memória obtendo **32**, seguindo pelo barramento este chega até o *accumulator* tendo como resultado **032**.

**3° CICLO** - *Program counter* envia sinal, passando pela *instruction register*, *address register* e chegando no *accumulator* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **3**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **2** a instrução **517**, no registro de instrução obtém-se o resultado **5** e no registro de endereço **17**, retornando até a casa **17** da memória obtendo **32**, seguindo pelo barramento este chega até o *accumulator* tendo como resultado **032**.

**4° CICLO** - *Program counter* envia sinal, passando pela *instruction register*, *address register* e chegando no *accumulator* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **4**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **3** a instrução **902**, no

registro de instrução obtém-se o resultado **9** e no registro de endereço **02**, por fim retornando o valor armazenado no acumulador **32** para o *output*.

**5° CICLO** - *Program counter* envia sinal, passando pela *intruction register*, *address register* e chegando no *acumulador* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **5**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **4** a instrução **514**, no registro de instrução obtém-se o resultado **5** e no registro de endereço **14**, retornando até a casa **14** da memória obtendo **32**, seguindo pelo barramento este chega até o *acumulador* tendo como resultado **032**.

**6° CICLO** - *Program counter* envia sinal, passando pela *intruction register*, *address register* e chegando no *acumulador* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **6**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **5** a instrução **922**, no registro de instrução obtém-se o resultado **9** e no registro de endereço **22**, por fim retornando o valor do conteúdo do acumulador **32** como um caractere para o *output*.

**7° CICLO** - *Program counter* envia sinal, passando pela *intruction register*, *address register* e chegando no *acumulador* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **7**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **6** a instrução **517**, no registro de instrução obtém-se o resultado **5** e no registro de endereço **17**, retornando até a casa **17** da memória obtendo **32**, seguindo pelo barramento este chega até o *acumulador* tendo como resultado **032**.

**8° CICLO** - *Program counter* envia sinal, passando pela *intruction register*, *address register* e chegando no *acumulador* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **8**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **7** a instrução **922**, no registro de instrução obtém-se o resultado **9** e no registro de endereço **22**, por fim retornando o valor do conteúdo do acumulador **32** como um caractere para o *output*.

**9° CICLO** - *Program counter* envia sinal, passando pela *instruction register*, *address register* e chegando no *accumulator* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **9**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **8** a instrução **115**, no registro de instrução obtém-se o resultado **1** e no registro de endereço **15**, retornando até a casa **15** da memória obtendo **01**, seguindo pelo barramento este chega até o *arithmetic unit* pegando o valor de *accumulator* **032** e somando o valor da casa **15 (+1)**, retornando como resultado **033** para armazenar no *accumulator*.

**10° CICLO** - *Program counter* envia sinal, passando pela *instruction register*, *address register* e chegando no *accumulator* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **10**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **9** a instrução **317**, no registro de instrução obtém-se o resultado **3** e no registro de endereço **17**, retornando até a casa **17** da memória, registra-se o valor do *accumulator* **033**.

**11° CICLO** - *Program counter* envia sinal, passando pela *instruction register*, *address register* e chegando no *accumulator* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **11**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **10** a instrução **216**, no registro de instrução obtém-se o resultado **2** e no registro de endereço **16**, retornando até a casa **16** da memória obtendo **97**, seguindo pelo barramento este chega até o *arithmetic unit* pegando o valor de *accumulator* **033** e subtraindo o valor da casa **16 (-97)**, retornando como resultado **-064** para armazenar no *accumulator*.

**12° CICLO** - *Program counter* envia sinal, passando pela *instruction register*, *address register* e chegando no *accumulator* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **12**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **11** a instrução **713**, no registro de instrução obtém-se o resultado **7** e no registro de endereço **13**. A instrução é decodificada e por meio do barramento de endereços, mas não em decorrência do valor do *accumulator* não ser zero, não é realizado o desvio para a posição **13** – Branch (if zero) to the address 13.

**13° CICLO** - *Program counter* envia sinal, passando pela *intruction register*, *address register* e chegando no *acumulator* se divide para *Arithmetic unit* tendo resultado **+1** e retorna para *program counter* tendo como total **13**, o segundo sinal percorre o barramento de saída para memória RAM recolhendo na casa **12** a instrução **602**, no registro de instrução obtém-se o resultado **6** e no registro de endereço **02**. A instrução é decodificada e por meio do barramento de endereços, é realizada uma instrução de pulo para as instruções da posição **2** – Branch to memory, address 2. Com isso, entra-se em um looping de operações até que o valor do *acumulator* seja **0**, permitindo-se que seja realizado branch para a posição **13**.