

O objetivo desta atividade utilizar as instruções lógicas, de deslocamento e rotações.

### Parte 1

Fazer um programa que leia um número binário do teclado. Código menor possível.

- Entrada de números binários:
  - *string* de caracteres "0's" e "1's" fornecidos pelo teclado;
  - CR é o marcador de fim de *string*;
  - BX é assumido como registrador de armazenamento;
  - máximo de 16 bits de entrada.
- Algoritmo básico em linguagem de alto nível:
  - Limpar BX
  - Entrar com um caractere "0" ou "1"
  - WHILE caractere diferente de CR DO
    - Converte caractere para valor binário
    - Deslocar BX 1 casa para a esquerda
    - Inserir o valor binário lido no LSB de BX
    - Entra novo caractere
  - END\_WHILE

Entregar arquivo lab08\_1.asm

### Parte 2

Fazer um programa que imprima um número binário. Código menor possível.

- Saída de números binários:
  - BX é assumido como registrador de armazenamento;
  - total de 16 bits de saída;
  - *string* de caracteres "0's" e "1's" é exibido no monitor de vídeo.
- Algoritmo básico em linguagem de alto nível:
  - FOR 16 vezes DO
    - rotação de BX à esquerda 1 casa binária (MSB vai para o CF)
    - IF CF = 1
      - THEN exibir no monitor caracter "1"
      - ELSE exibir no monitor caracter "0"
  - END\_IF
  - END\_FOR

Entregar arquivo lab08\_2.asm

### Parte 3

Fazer um programa que leia um número hexadecimal do teclado. Código menor possível.

- Entrada de números hexadecimais:
  - BX é assumido como registrador de armazenamento;
  - *string* de caracteres "0" a "9" ou de "A" a "F", digitado no teclado;
  - máximo de 16 bits de entrada ou máximo de 4 dígitos hexa.
- Algoritmo básico em linguagem de alto nível:
  - Inicializa BX
  - Entra um caractere hexa
  - WHILE caractere diferente de CR DO
    - Converte caractere para binário
    - Desloca BX 4 casas para a esquerda
    - Insere valor binário nos 4 bits inferiores de BX
    - Entra novo caractere
  - END\_WHILE

Entregar arquivo lab08\_3.asm

### Parte 4

Fazer um programa que imprima um número hexadecimal. Código menor possível.

- Saída de números hexadecimais:
  - BX é assumido como registrador de armazenamento;
  - total de 16 bits de saída;
  - *string* de caracteres HEXA é exibido no monitor de vídeo.
  -
- Algoritmo básico em linguagem de alto nível:
  - FOR 4 vezes DO
    - Mover BH para DL
    - Deslocar DL 4 casas para a direita
    - IF DL < 10
      - THEN converte para caractere na faixa 0 a 9
      - ELSE converte para caractere na faixa A a F
  - END\_IF

Exibição do caractere no monitor de vídeo  
Rodar BX 4 casas à esquerda

END\_FOR  
Entregar arquivo lab08\_4.asm

#### **Parte 4**

Fazer um programa que tenha 4 procedimentos, com os programas das partes anteriores, e permita ao usuário escolher qual entrada e qual saída. Código menor possível.