

O objetivo desta atividade é permitir que o aluno seja capaz de utilizar as instruções lógicas, de deslocamento, multiplicação e divisão

### 1- Entrada binária

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:

```
Limpa BX
Entra um caracter "0" ou "1"
WHILE caracter diferente de CR DO
    Converte caracter para valor binário
    Desloca BX 1 casa para a esquerda
    Insere o valor binário lido no LSB de BX
    Entra novo caracter
END_WHILE
```

### 2- Saída binária

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
```

### 3- Entrada hexadecimal

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 caracter hexa
WHILE caracter diferente de CR DO
    Converte caracter para binário
    Desloca BX 4 casas para a esquerda
    Insere valor binário nos 4 bits inferiores de BX
    Entra novo caracter
END_WHILE
```

#### 4- Saída hexadecimal

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 caracter na faixa 0 a 9
        ELSE converte para caracter na faixa A a F
    END_IF
    Exibição do caracter no monitor de vídeo
    Rodar BX 4 casas à esquerda
END_FOR
```

#### 5- Entrada decimal

Algoritmo básico em linguagem de alto nível:

```
total = 0
negativo = FALSO
ler um caractere
CASE caractere IS
    ' - ' : negativo = VERDADEIRO e ler um caractere
    ' + ' : ler um caractere
END_CASE
REPEAT
    converter caractere em valor binário
    total = 10 x total + valor binário
    ler um caractere
UNTIL caractere é um carriage return (CR)
```

```
IF negativo = VERDADEIRO  
    THEN total = - (total)  
END_IF
```

#### **6- Saída decimal**

Algoritmo básico em linguagem de alto nível:

```
IF AX < 0  
    THEN  exibe um sinal de menos substitui-se AX pelo seu complemento de 2  
END_IF  
contador = 0  
REPEAT  
    dividir quociente por 10  
    colocar o resto na pilha  
    contador = contador + 1  
UNTIL  quociente = 0  
FOR contador vezes DO  
    retirar um resto (número) da pilha  
    converter para caracter ASCII  
    exibir o caracter no monitor  
END_FOR
```

#### **Entregas**

Fazer um programa que permita a entrada e qualquer uma das bases e a saída em qualquer uma das bases. O programa deverá perguntar em que base será a entrada do número e em que base será a saída do número.