

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

#### TAREFA ÚNICA

##### 1- Entrega

Fazer um programa que permita a entrada e qualquer uma das bases e a saída em qualquer uma das bases, ou seja, faz a conversão de bases. O programa deverá perguntar em que base será a entrada do número e em que base será a saída do número. Usar procedimentos. As bases são: binária, decimal e hexadecimal. Abaixo os algoritmos que vem ser codificados em assembly. Cada um deste algoritmos deve ser um procedimento. Não esquecer dos comentários e de salvar, quando possível os registradores na pilha, de modo a permitir sua utilização por qualquer outro usuário, sem que ele conheça o algoritmo.

##### 2- 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 caractere "0" ou "1"
WHILE caractere diferente de CR DO
    Converte caractere para valor binário
    Desloca BX 1 casa para a esquerda
    Insere o valor binário lido no LSB de BX
    Entra novo caractere
END_WHILE
```

##### 3- 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 caractere "1"
        ELSE exibir no monitor caractere "0"
    END_IF
END_FOR
```

#### 4- 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 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
```

#### 5- 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 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
```

#### 6- 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
```

### 7- 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 caractere ASCII
    exibir o caractere no monitor
END_FOR
```

### Entrega

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. Usar procedimentos e macros.