

Meu assembler ignora comentários, ou seja, quando lê ‘;’, ele ignora a linha toda restante, mas não ignora o que vem antes do ‘;’.

.DATA representa os valores:

1. Os valores podem assumir um intervalo de -128 até 127. Somente números inteiros.
2. Deve sempre conter uma letra sem valor definido, para servir de resposta final (R ou X, por exemplo).
3. No caso de um assembly com multiplicação, ele deve conter duas letras para servir de auxílio com os valores de 1 e -1, necessários para realizar a multiplicação no Neander.

.CODE contém a lógica do programa para realizar o cálculo.

1. O “.ORG 0” não está funcional, é apenas visual.
2. Só deixei de implementar os mnemônicos de “OR” e “AND”, pois não achei casos onde eles seriam necessários para as contas. Todos os outros funcionam perfeitamente, inclusive os Jumps.
3. Os jumps funcionam pulando para a linha indicada após o código, e a indexação de cada linha começa por 0 a partir da primeira linha de CÓDIGO, ou seja, após o “.ORG 0”, e vai incrementando a cada nova linha.