

Teste 2 – Recursão

Em algumas situações muito específicas, utiliza-se o formato chamado *Decimal Zonado* para representar números decimais. Podemos simulá-lo por meio de uma *string* onde um dígito é representado por um par de caracteres, formado pela letra F (1º caractere) seguida por um caractere referente a um dos 10 dígitos decimais (2º caractere). A representação decimal zonada do inteiro 1805 é F1F8F0F5. Considere que os números têm no máximo 5 dígitos numéricos.

Escreva uma função **recursiva** que recebe uma *string* com pelo menos dois caracteres, representando um número inteiro no formato decimal zonado e retorne o número inteiro (int) correspondente. Por exemplo, se a *string* recebida pela função é F1F8F0F5, então o inteiro retornado é 1805.

Escreva também uma outra função **recursiva** que faz a operação ao contrário, isto é: converte um número inteiro numa *string* no formato decimal zonado. Para facilitar, considere o seguinte protótipo:

```
char * nomeDeSuaFuncao(int , char * );
```

onde a função recebe um número inteiro e o endereço de uma área de memória suficiente para uma *string* de 10 dígitos (mais o terminador) e coloca nesta área o inteiro no formato decimal zonado. Esta função também retorna o endereço do terminador da *string*.

Teste obrigatoriamente para números diferentes dos usados nos exemplos acima. Para testar a segunda função, faça a alocação da memória dinamicamente. Para testar se a segunda função retorna a posição do terminador, imprima a diferença entre esta posição e o início da *string* (o que deve dar o comprimento da *string*).

Não é permitido o uso das funções de conversão entre caracteres e inteiros (*atoi()* e *itoa()*).

Dica1: para a primeira função, observe que basta ir somando $digito \times 10^{posicao}$.

Dica2: para a segunda função, basta ir dividindo por 10, enquanto o número é maior do que 10. Como a função retorna a posição do terminador, ao chegar a um número menor que 10, basta colocar a informação a partir do terminador (e escrever por cima dele).