



UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI

ALGORITMOS E ESTRUTURAS DE DADOS I E INTRODUÇÃO A CIENCIA DA COMPUTAÇÃO

Alunos: Arthur Gabriel e Lucas Grandolpho

Curso: Ciência da Computação

Professores: Elverton Fazzion e Carol

São João del Rei, maio de 2018

1-Introdução

O algoritmo em questão procura resolver um simples problema de conversão de bases, no qual o indivíduo digita um número qualquer, a base que ele está, e a base a ser convertida. O programa irá retornar o número convertido para a base desejada sem problemas, deixando o processo de conversão de bases um processo automatizado pelo computador facilitando a vida dos futuros usuários. O programa atual se limita a conversão das seguintes bases: BINARIA, OCTAL, DECIMAL, HEXADECIMAL.

2-Solução do Problema

O código começa pedindo para que o usuário entre com um número iniciado com o sinal e com a parte inteira separada da parte fracionário por um ".", após isso espera-se um "space" para colocar a base atual e outro "space" para colocar a base desejada.

Exemplo: +101.01 2 10

A string de caracteres será lida e comparada com a saída padrão (+0 0 0), e caso as strings forem iguais o programa quebra o comando de repetição para a entrada de valores e passa para a conversão deles. Caso seja diferente a string será tratada e transformada em um número na base DECIMAL dentro da função 'TrataNum()'.

Dentro da função 'TrataNum()':

A função irá separar os elementos da String 'n_orig[]' onde se localiza o número. Primeiro ela irá pegar o valor da base atual e guardar na variável inteira bases[0], e depois guardar o valor da base a ser convertida na variável bases[1].

Depois a função irá fazer um somatório com os valores da parte inteira, de forma que, independente de sua base, a parte inteira será convertida para a base DECIMAL e guardada na variável double 'soma'.

De forma similar, a função irá fazer outro somatório com os valores da parte fracionária, de forma que, independente de sua base, a parte fracionária será convertida para a base DECIMAL e guardada na variável double 'soma'.

Após a conversão do número para decimal, a função irá converter o sinal, ou seja, caso o número de entrada fosse negativo o valor da variável 'soma' será multiplicado por (-1) . Caso contrário, a função implementa o valor de 'soma' dentro da variável global double 'numeros[][0]' e 'bases[1]' em 'numeros[][1]'.

Após o termino da função 'TrataNum()' o programa irá pedir para o usuário digitar outro número até que a entrada +0 0 0 e o programa passe a converter os valores digitados para as bases desejadas.

Nesta parte o programa irá executar a função 'converteNum()'.

Dentro da função 'converteNum()':

Primeiramente a função irá reconhecer o sinal (+ || -) do número e imprimir o sinal na tela. Após isso, o algoritmo irá separar a parte inteira da parte fracionária. Lembrando que, o número está com seu valor na base DECIMAL.

Dessa forma é feito um somatório para a parte inteira e outro para a parte fracionária, de forma que:

→Parte inteira: O número recebe o somatório $\sum_{i=1}^n a_i \times b^{i-1}$

e já imprime o algarismo.

O programa imprime um "." marcando o final da parte inteira.

→Parte Fracionária: O número recebe o somatório $\sum_{i=1}^2 a_{i_{10}} \times 16_{10}^{(i-1)_{10}}$

e já imprime o algarismo.

Após isso o algoritmo quebra uma linha e começa a conversão de outro número e assim por diante até o +0 0 0.

3-Testes Realizados

Os testes realizados comprovam a veracidade do programa:

```
Digite a entrada da forma <NUMERO> <BASE ORIGEM> <BASE ALVO>: (Entre com "+0 0 0" para encerrar a entrada)
+160. 10 16
-101.01 2 10
+15. 10 16
-14.625 10 8
+2018.01 8 2
+EA.AE 16 10
+100.01 10 2
-10101.1001 2 8
+0 0 0
-----
+A0.
-5.25
+F.
-16.5
+10000010000.000001
+234.6796875
+1100100.00000010100011110101110000101
-25.44

Process returned 0 (0x0)   execution time : 25.555 s
Press any key to continue.
```

OBS: Os testes foram realizados em um windows 10. Intel core I5, 2.3Ghz, 8GB RAM.