

Linux Line Calculator

Problem A

O objetivo é escrever um programa simples que obedece o sintaxe padrão de Linux. O programa deverá ler linhas de texto do input padrão (stdin) ou dum ou mais ficheiros e fazer o calculo (soma ou multiplicação) dos números contidas na linha fazendo o output no mesmo base.

The objective of this exercise is to create a simple command that follows the Linux standard. The program will read lines of text from the standard input and calculate either a sum or multiplication of the numbers contained on the line of text.

Task - PT

O objetivo é escrever um programa que faça uma operação matemática com um conjunto de números numa ou mais linhas de texto.

A sintaxe do comando, que chamaremos linecalc, é:

```
linecalc [ -base=value ] [-op=value] [ficheiros]
```

As opções validas são: -base=10 -base=8 ou -base=16 e -op=+ ou -op=*

Por defeito o comando fará a operação de soma (+) com números inteiros de base 10.

Em todos os casos é a ultima opção que conta. Quer dizer se existem múltiplas bases é apenas a ultima que conta.

Por defeito, o comando faz a leitura no "standard input" exceto quando os últimos argumentos são ficheiros que serão abertas e lidas um após o outro.

Task -EN

Write a program to execute a mathematical operation, sum or multiply, on a list of numbers.

The syntax of the command, linecalc, is:

```
linecalc [ -base=value ] [-op=value] [ficheiros]
```

Valid options are -base=10 -base=8 or -base=16 and -op=+ or -op=*

By default the command will execute the sum operation over the list of integer numbers in base 10.

In case of multiple options of the same type it is always the last option in the list that matters.

By default the command reads from the standard input except when the last arguments are filenames which are opened one after another.

Input

O input consiste em texto lido pelo standard input ou a partir dum ou mais ficheiros cujo nomes são passados como argumentos. Linhas do texto contêm números inteiros hexadecimais, decimais ou octais separados por um espaço. Linhas terminam com um \n

The input is text either from the standard input or from files. Lines of text contain either hexadecimal values, decimals or octal values separated by a space. Lines are terminated with a \n.

Output

Para cada linha de texto o valor do output é o valor do calculo matematico (somar ou multiplicar). Caso que um ficheiro de input não existe o programa deverá emitir (no stdout) a mensagem "File Not Found fileName"

For each line of input the output is the result of the mathematical operation (sum or multiply). If any of the files specified do not exist the program must output the message "File Not Found fileName" where filename is the name of the file not found.

Constraints

O número de caracteres numa linha de texto é sempre menor ou igual a 1000.

Todo o output tem que ser no canal de stdout.

O programa deverá sempre devolver o valor zero (return 0)

The number of characters in a line of text is less than or equal to 1000.

All output is on the stdout channel (even errors)

The program MUST always terminate with return 0 or exit(EXIT_SUCCESS)

Samples

Sample Input 1

Considere o ficheiro in1.txt com as duas linhas de texto.

Consider the file in1.txt that contains the following two lines of text.

```
1 2 3 4 5
6 7 8
```

Sample Executions 1 with Output

```
linecalc.exe < in1.txt
15
21
linecalc in1.txt
15
21
linecalc in1.txt in1.txt
15
21
15
21
linecalc -base=10 in1.txt
15
21
linecalc -base=8 in1.txt
17
24
linecalc -base=16 in1.txt
f
15
linecalc -base=10 -op=* in1.txt in1.txt
120
```

```
336
linecalc -base=16 -op=* in1.txt
78
150
linecalc -base=16 -base=10 -op=* in1.txt
120
336
```

Sample Execution 2

```
linecalc lixo
File Not Found lixo
```

Sample Execution 3

```
linecalc -base=16
```

```
ab a ↵
b5
ab 5 ↵
b0
```

Useful C Functions

getline() and fgets()
strtok() string to token
sscanf read from string
For decimal, octal and hexadecimal formats
%d, %o and %x format qualifiers
strlen()

ALL COMPILATION must be with the ALL WARNINGS flag
cc -Wall

Before submitting a solution there must be no warnings from the compiler.
see <https://gcc.gnu.org/onlinedocs/gcc/Warning-Options.html>

hints

Read lines of text to a string

Parse a line string into little strings using tokenizer strtok() .. click aqui <http://www.br-c.org/doku.php?id=strtok>

read the little string using sscanf() .. which is scanf from a string

A função sscanf é idêntica à função scanf, mas os dados são lidos da string.