

# Projeto “cálculo de uma expressão matemática”

## Descrição

Nas aulas da disciplina de PCII, você viu: a) como verificar se uma expressão é “bem formada”; b) como calcular o resultado de uma expressão escrita na Notação Polonesa Reversa. Além disso, o professor mais bonzinho que você já conheceu até hoje, até disponibilizou um programa capaz de converter uma expressão escrita na notação infixa para a notação posfixa.

No entanto, você pensou com seus “botões”: se o compilador gcc consegue calcular o resultado de uma expressão qualquer, por que eu também não posso fazer um programa capaz de fazer isso? Afinal, sou um aluno do curso de SI e não vai ser um compilador qualquer que vai fazer algo que eu não possa fazer igual ou até melhor!

Com essa idéia fixa em mente você parte ansiosamente para a construção desse programa. E definiu que o seu programa deve ser capaz atender as seguintes condições mínimas:

- Ler da entrada padrão uma expressão qualquer!
- Essa expressão deve poder conter números inteiros, os caracteres de abre e fecha parênteses e os 5 sinais algébricos: ‘+’, ‘-’, ‘/’, ‘\*’, ‘\$’ (exponenciação);
- Os números inteiros componentes da expressão podem ser qualquer número no intervalo [0; 32767];
- A expressão de entrada estará sempre escrita na notação infixa;
- Como o usuário pode esquecer algum abre e/ou fecha parênteses, o programa deverá sempre verificar, antes de fazer qualquer cálculo, se a expressão lida de *stdin* é uma expressão “bem formada”;
- Uma vez lida a expressão de *stdin*, seu programa deve mostrar na tela o resultado do cálculo dessa expressão.

A seguir veja alguns exemplos de entradas e saídas que o seu programa deve ser capaz de calcular:

Entrada padrão	Saída
$25 + 32 + (5 * 2)$	67
$32000 - 30000 + (3 \$ 2) - 9$	2000
$(( (3 - 2) * (5 - 3) ) \$ 2 + (4 - 6)) / 2$	1