

**Prova final de Programação de Computadores - CI-208A Prof. Marcos  
Alexandre Castilho 17 de julho de 1995.**

Suponha que desejemos escrever um programa para adicionar dois polinômios, por exemplo,

$$(1 + 2x + 3x^3) + (2 - x) = 3 + x - 3x^3$$

Em geral, suponha que o programa deve computar  $r(x) = p(x) + q(x)$ , onde  $p$  e  $q$  são polinômios com  $N$  coeficientes.

Neste programa, o polinômio  $p(x) = p_0 + p_1x + \dots + p_{N-1}x^{N-1}$  está representado pelo arranjo (vetor)  $p[0..N-1]$ , sendo que  $p[j] = p_j$ . Um polinômio de grau  $N-1$  é definido por  $N$  coeficientes. Em C, o tamanho máximo de  $N$  tem que ser previsto; Evidentemente  $N_{max}$  tem que ser definido para o grau máximo que estejamos antecipando. Isto é inconveniente se o programa for utilizado várias vezes com diferentes valores de  $N$  que possam variar muito.

Faça um programa em C, que:

1. Defina uma constante `Nmax` com valor 100
2. Defina o tipo `POLINOMIO` como sendo um vetor de `Nmax` posições
3. Faça uma função em C que leia os coeficientes e expoentes (dois em cada linha) de um polinômio, e os insira na posição adequada do arranjo:  
**void LePoli (POLINOMIO P);**
4. Faça uma função em C que some dois polinômios:  
**void SomaPoli (POLINOMIO P, POLINOMIO Q, POLINOMIO SOMA);**
5. Faça uma função em C que multiplique dois polinômios:  
**void MultiplicaPoli (POLINOMIO P, POLINOMIO Q, POLINOMIO MULT);**
6. Faça uma função em C que imprima um polinômio:  
**void ImprimePoli (POLINOMIO P);**

As funções acima devem ser feitas de tal modo que o seguinte programa abaixo funcione normalmente (a menos dos *include's* necessários):

---

```
main ()
{
    POLINOMIO p, q, r, m;

    LePoli (p); imprimePoli (p);
    LePoli (q); imprimePoli (q);
    PoliAdd (p, q, r); imprimePoli (r);
    PoliMult (p, q, m); imprimePoli (m) ;
}
```

---