

EXERCÍCIO PROGRAMA 2

Data de entrega: 25/10/2001

Problema. Implemente em C uma calculadora simples que manipula números inteiros de tamanho arbitrário. A sua calculadora deve ser capaz de executar as seguintes operações, no mínimo:

1. somar dois inteiros,
2. calcular a diferença de dois inteiros,
3. calcular o produto de dois inteiros,
4. calcular o quociente inteiro de dois inteiros,
5. calcular o resto da divisão de um inteiro por outro.

A sua calculadora deve trabalhar com a notação polonesa reversa (como calculadoras HP). Você também pode implementar memórias, para tornar sua calculadora mais fácil de usar. A sugestão natural é você implementar cada número como uma lista ligada.

Seja inventivo! Incremente sua calculadora implementando mais coisas! Por exemplo, para fazer sua calculadora ser realmente útil, você poderia implementar uma *calculadora programável* (isto é, uma máquina que permite escrever pequenos programas).

Sugestão. Você pode achar úteis os seguintes programas do Sedgewick (Algorithms in C, Capítulo 4).

```
-----
#include <stdio.h>
#include <stdlib.h>
#include "POLY.h"
main(int argc, char *argv[])
{ int N = atoi(argv[1]); float p = atof(argv[2]);
  Poly t, x; int i, j;
  printf("Binomial coefficients\n");
  t = POLYadd(POLYterm(1, 1), POLYterm(1, 0));
  for (i = 0, x = t; i < N; i++)
    { x = POLYmult(t, x); showPOLY(x); }
  printf("%f\n", POLYeval(x, p));
}
-----

typedef struct poly *Poly;
void showPOLY(Poly);
Poly POLYterm(int, int);
Poly POLYadd(Poly, Poly);
Poly POLYmult(Poly, Poly);
float POLYeval(Poly, float);
-----
```

```
#include <stdlib.h>
#include "POLY.h"
struct poly { int N; int *a; };
Poly POLYterm(int coeff, int exp)
{ int i; Poly t = malloc(sizeof *t);
  t->a = malloc((exp+1)*sizeof(int));
  t->N = exp+1; t->a[exp] = coeff;
  for (i = 0; i < exp; i++) t->a[i] = 0;
  return t;
}
Poly POLYadd(Poly p, Poly q)
{ int i; Poly t;
  if (p->N < q->N) { t = p; p = q; q = t; }
  for (i = 0; i < q->N; i++) p->a[i] += q->a[i];
  return p;
}
Poly POLYmult(Poly p, Poly q)
{ int i, j;
  Poly t = POLYterm(0, (p->N-1)+(q->N-1));
  for (i = 0; i < p->N; i++)
    for (j = 0; j < q->N; j++)
      t->a[i+j] += p->a[i]*q->a[j];
  return t;
}
float POLYeval(Poly p, float x)
{ int i; double t = 0.0;
  for (i = p->N-1; i >= 0; i--)
    t = t*x + p->a[i];
  return t;
}
```