

## CI2126 QUIZ 1 5%

Dado el código de la página siguiente que representa un TAD Polinomio, implemente las operaciones

- anyadirTerm*
- sumaPoli*;

Se recibe la siguiente entrada

```
1.0 5   -1.0 3   1.0 1   0.0 -1
2.0 4   -3.0 2   -1.0 1   1.0 0   0.0 -1
```

y la salida debe ser

```
+ 1.0 x^5 + 2.0 x^4 - 1.0 x^3 - 3.0 x^2 + 1.0 x^0
```

BONO:

¿Que modificaciones hay que hacerle a *sumaPoli* para implementar *multPoli* ? (Sólo indicar)

```
#include <stdio.h>
#include <stdlib.h>

const int MAX = 10;

typedef struct Poli_tag {
    int gradoMayor;
    float* coef;
} Poli_t;

typedef Poli_t* Poli; /* El apuntador a la anterior */

Poli consPoli (int maxGrado) {
    _pre( maxGrado > 0 );
    Poli p = (Poli) malloc(sizeof(Poli_t));
    p->coef = (float*) calloc(maxGrado+1, sizeof(float));
    p->gradoMayor = maxGrado;
    return p;
}

void anyadirTerm(Poli p, float coef, int grado); {
    _pre( ( p != NULL) and (grado <= p->gradoMayor) );
    p->coef[grado] = coef;
}
```

```
Poli sumaPoli( Poli a, Poli b); {
    _pre( ( a != NULL) and (b != NULL) );

    Poli s = consPoli(MAX_GRADO);
    int k;
    for (k=MAX(a->gradoMayor, b->gradoMayor); k > -1; k--)
        s->coef[k] = a->coef[k] + b->coef[k];
    return s;
}

void leerPoli ( Poli p) {
    _pre( p != NULL );
    int grado;
    float coef;
    scanf("%f", &coef);
    scanf("%i", &grado);
    printf("Poli: ");
    while (grado >=0) {
        anyadirTerm(p, coef, grado);
        scanf("%f", &coef);
        scanf("%i", &grado);
    }
}

void escribirPoli (Poli p) {
    _pre( p != NULL );
    int k;
    for (k = p->gradoMayor; k > -1; k--)
        if (p->coef[k] > 0.0)
            printf(" + %f x^%i", p->coef[k], k);
        else if (p->coef[k] < 0.0)
            printf(" - %f x^%i", -p->coef[k], k);
}

int main () {
    Poli x = consPoli(MAX);
    Poli y = consPoli(MAX);
    Poli z = consPoli(MAX);
    leerPoli(x);
    leerPoli(y);
    z = sumaPoli(x,y);
    escribirPoli(z);
    return 0;
}
```