

## 1. Introducción

Este programa permite calcular los coeficientes de un polinomio derivado y cambiado de base. Para ello hay que introducir en el programa los coeficientes de un polinomio y el punto para hacer el cambio de base.

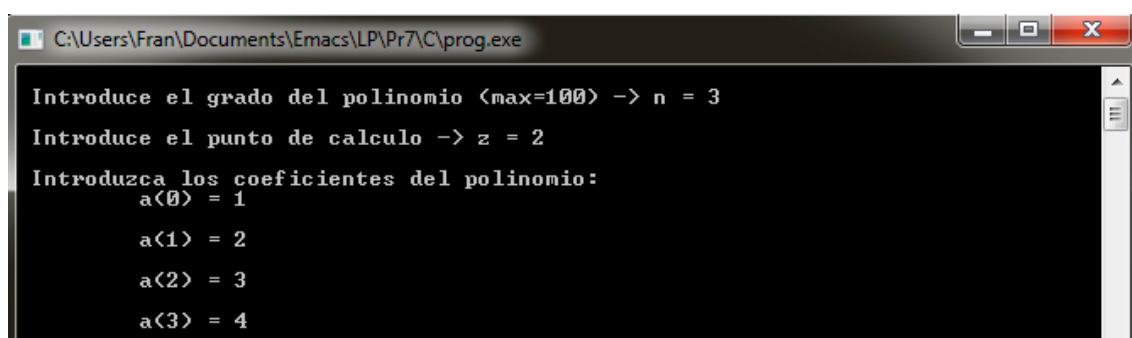
Este programa se ha traducido de un programa ya existente de características similares escritas en FORTRAN 77.

## 2. Manejo del programa

Vamos demostrar cómo usar el programa con un ejemplo sencillo. Cogemos el polinomio  $P_n(x) = 1 + 2x + 3x^2 + 4x^3$  y calcularemos los coeficientes resultantes de derivar el polinomio hasta que no queden incógnitas (es decir hasta que el polinomio valga 0). Pero lo haremos para el punto  $z = 2$ .

- Introducir polinomio y punto para el cambio de base:

Introducimos primero el grado del polinomio (en nuestro ejemplo 3). Luego el punto para el cambio de base, si no desea cambio de base introduzca 0. Para nuestro ejemplo cambiaremos al punto  $z = 2$ . Luego introducimos los coeficientes del polinomio uno por uno.



```
C:\Users\Fran\Documents\Emacs\LP\Pr7\C\prog.exe

Introduce el grado del polinomio <max=100> -> n = 3
Introduce el punto de calculo -> z = 2
Introduzca los coeficientes del polinomio:
  a<0> = 1
  a<1> = 2
  a<2> = 3
  a<3> = 4
```

- Interpretar resultados:

Lo primero vamos a realizar el cambio de base a mano. El polinomio  $P_n(x) = 1 + 2x + 3x^2 + 4x^3$  para  $z = 2$  vale  $P_n(x + 2) = 1 + 2(x + 2) + 3(x + 2)^2 + 4(x + 2)^3$ . Desarrollando el polinomio obtenemos  $P_n(x) = 49 + 62x + 27x^3 + 4x^3$ . Finalmente calculamos los coeficientes de derivar sucesivas veces (ver tabla). El programa obtiene lo mismo resultados que nosotros a mano en menos tiempo.

Coeficiente	Calculo	Resultado
$P_0$	49	49
$P_1$	$62 * 1$	62
$P_2$	$27 * 3 * 2 * 1$	54
$P_3$	$4 * 3 * 2 * 1$	24

```
C:\Users\Fran\Documents\Emacs\LP\Pr7\C\prog.exe

a<1> = 2
a<2> = 3
a<3> = 4

RESULTADOS
Grado del polinomio -> n =      3
Punto de calculo -> z =      2.0000000000
Coeficiente en la base {1,x,...,x^n}:
  a[0] =      1.0000000000
  a[1] =      2.0000000000
  a[2] =      3.0000000000
  a[3] =      4.0000000000
Coeficiente en la base {1,(x-z),...,(x-z)^n}:
  a[0] =     49.0000000000
  a[1] =     62.0000000000
  a[2] =     27.0000000000
  a[3] =      4.0000000000
Funcion y derivadas en el punto x = z:
  a[0] =     49.0000000000
  a[1] =     62.0000000000
  a[2] =     54.0000000000
  a[3] =     24.0000000000
```

### 3. Comentarios

Adjunto al final del documento el código, para compilarlo aconsejo usar el archivo punto bat que se encuentra en el link. Ya que a diferencia de otros programas hechos por mi este tiene las funciones en archivos separados.

Adjunto también un hyperlink para mayor comodidad por si se quiere descargar el programa.

### 4. Link



<https://onedrive.live.com/redir?resid=545e57c2bf8bfc8b!303&authkey=!AB49pzmPPIVl4pY&ithint=folder%2cc>

```

/*
PROGRAMA PARA DERIVAR UN POLINOMIO

1. El usuario introduce los datos del polinomio, esto se hace en el metodo leerCoeficientes()
Polinomio que sigue esta forma:
P_n(x) = a_0 + a_1*x + a_2*x^2 + a_3*x^3 + ... + a_n*x^n
Donde n es el grado del polinomio.

2. Luego el usuario introduce el punto z.
Se realiza el cambio de base a la base (x-z). Para ello se divide el polinomio por (x-z).
Esto se realiza en el metodo divisionSintetica().

3. Se calcula las derivadas
      (i
      d P_n(x)
c (i) = -----
      (i
      d x      | x = z
*/

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

#define MAX 100

int main (int argc, char *argv[]) {
    /*prototype*/
    void leerCoeficientes(int, double *);
    void divisionSintetica(int, double, double *, double *);
    void calcularDerivadas(int, double *, double *);
    void escribirResultados(int, double, double *, double *, double *);
    /*variables*/
    int n; //polinomial's grade

    do{
        printf("\n Introduce el grado del polinomio (max=100) -> n = ");
        scanf("%d", &n);
    } while (n <= 0 || n >= 100);
    double a[n];
    double b[n];
    double c[n];

    double z;
    printf("\n Introduce el punto de calculo -> z = ");
    scanf("%lf", &z);

    /*call methods*/
    leerCoeficientes(n,a);
    divisionSintetica(n,z,a,b);
    calcularDerivadas(n,b,c);
    escribirResultados(n,z,a,b,c);

    /*wait for user keyboard*/
    fflush(stdin); getchar();

    return 0;
} //end of method main

```

```
#include <stdio.h>
#include <stdlib.h>
/*metodo para leer de la pantalla los coeficientes introducidos por el usuario*/
void leerCoeficientes(int n, double v[]){

    printf("\n Introduzca los coeficientes del polinomio: ");
    int i;
    for(i=0; i <= n; i++){
        printf("\n\t a(%d) = ",i);
        scanf("%lf",&v[i]);
    }
    return ;

}

} //end of method leerCoeficientes
```

Función LeerCoeficientes

```
#include <stdio.h>
#include <stdlib.h>
/*metodo para dividir el polinomio por (x-z)*/
void divisionSintetica(int n, double z, double a[], double b[]){
    int i;
    for(i = 0; i <= n; i++)
        b[i] = a[i];

    int k;
    for(k = n; k >= 1; k--){
        for(i = n-1; i >= n-k; i--)
            b[i] += b[i+1]*z;
    }

    return ;

}

}
```

Función DivisiónSintetica

```
#include <stdio.h>
#include <stdlib.h>
/*metodo para dividir el polinomio por (x-z)*/
void calcularDerivadas(int n, double a[], double b[]){
    int i;
    for(i = 0; i <= n; i++)
        b[i] = a[i];

    double fact = 1.;
    for(i = 2; i <= n; i++){
        fact *= (double)i;
        b[i] *= fact;
    }

    return ;

}

}
```

Función CalcularDerivadas

```

#include <stdio.h>
#include <stdlib.h>
/*metodo para dividir el polinomio por (x-z)*/
void escribirResultados(int n, double z, double a[], double b[], double c[]){
    printf("\n RESULTADOS");

    printf("\n Grado del polinomio -> n = %5d",n);
    printf("\n Punto de calculo -> z = %20.9f",z);

    printf("\n Coeficiente en la base {1,x,...,x^n}:");
    int i;
    for( i = 0; i <= n; i++){
        printf("\n\ta[%d] = %20.9f",i,a[i]);
    }

    printf("\n Coeficiente en la base {1,(x-z),..., (x-z)^n}:");
    for( i = 0; i <= n; i++){
        printf("\n\ta[%d] = %20.9f",i,b[i]);
    }

    printf("\n Funcion y derivadas en el punto x = z:");
    for( i = 0; i <= n; i++){
        printf("\n\ta[%d] = %20.9f",i,c[i]);
    }

    return ;
}

```