



UNIVERSIDAD
DE MÁLAGA

| uma.es

Dpto. de Lenguajes y Ciencias de la
Computación

Programación de Sistemas y
Concurrencia

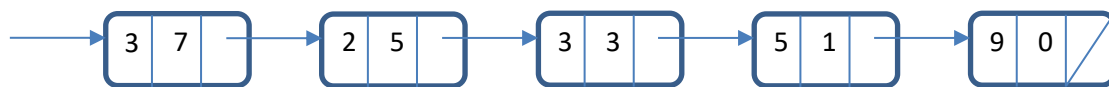
Primer Control C
Curso 2019-2020

APELLIDOS _____ NOMBRE _____

DNI _____ ORDENADOR _____ GRUPO/TITULACIÓN _____

VERSIÓN 4. Bloque de C-Ejercicio Polinomios-Primera Parte

Un polinomio del tipo $3x^7 + 2x^5 + 3x^3 + 5x + 9$ podría representarse mediante una lista enlazada del modo siguiente:



Cada nodo contiene un monomio del polinomio, con su **coeficiente**, su **exponente**, y un enlace al monomio siguiente. Como puedes observar en el dibujo los monomios están ordenados según su exponente, de mayor a menor (desde el monomio con exponente mayor ($3x^7$) al monomio con exponente menor (9)). Observa, asimismo, que los monomios que tienen coeficiente 0 **no** están almacenados en la lista (por ejemplo, el monomio de exponente 6 no está en la lista porque su coeficiente es 0).

En esta **primera parte** del examen, debes implementar las siguientes funciones en el fichero fuente Polinomio.c. Se proporciona un fichero Principal.c para que puedas probar las funciones que vas implementando.

```
/*Crea el polinomio 0 (es decir, un polinomio vacío).
*/
void polinomioCero(TPolinomio *p);

/*Devuelve el grado del polinomio, es decir, el mayor exponente de los
monomios que no son nulos. En el ejemplo, el grado es 7. El grado del
polinomio cero es 0.
*/
unsigned int grado(TPolinomio p);

/* Devuelve el coeficiente de exponente exp del polinomio p.
*/
unsigned int coeficiente(TPolinomio p,unsigned int exp);

/* Inserta el monomio con coeficiente coef y exponente exp en el polinomio,
de manera que el polinomio quede ordenado. Asegurarse que no se insertan
monomios cuyo coeficiente sea 0 y tampoco dos monomios con el mismo
exponente. Si al insertar un monomio ya hay otro con el mismo exponente los
coeficientes se sumarán. El coeficiente y el exponente son números
naturales (enteros no negativo) */
void insertar(TPolinomio *p,unsigned int coef,unsigned int exp);
```

```

/*Escribe por la pantalla el polinomio con un formato similar al siguiente:
[(3,7)(0,6)(2,5)(0,4)(3,3)(0,2)(5,1)(9,0)] para el polinomio ejemplo.
Ten en cuenta que los monomios de exponente menor al grado del polinomio
con coeficiente 0 también aparecen en la salida.
*/
void imprimir(TPolinomio p);

/* Elimina todos los monomios del polinomio haciendo que el polinomio
resultante sea el polinomio 0.
*/
void destruir(TPolinomio *p);

```

Ejercicio Polinomios-Segunda Parte

Añade al fichero Polinomio.h el prototipo de la siguiente función, y añade su implementación al fichero Polinomio.c. En el fichero Principal.c quita los comentarios al código que se proporciona para que puedas probar esta función.

```

/* Lee los datos de un polinomio de un fichero de texto, y crea la lista de
monomios p. El formato del polinomio en el fichero contiene una secuencia
de pares de dígitos correspondientes al coeficiente y exponente de cada
monomio del polinomio, incluyendo los que tienen coeficiente nulo. En ambos
casos, suponemos que los coeficientes y exponentes son dígitos del 0 al 9
(no hay números superiores). Por ejemplo, para el polinomio de ejemplo el
fichero de texto estaría compuesto por la secuencia de caracteres
"0690332551370402". Observa que los monomios pueden venir desordenados en
el fichero de entrada. La conversión de un valor de tipo 'char' que
contenga un valor numérico (ej. char c = '2') a su correspondiente valor
entero (int valor), se puede hacer de la siguiente forma: valor = c - '0'
*/
int crearDeFichero(TPolinomio *p, char *nombre);

```

Ejercicio Polinomios-Tercera Parte

Implementa la siguiente función **en el fichero** Principal.c que se proporciona. Además, en el fichero Principal.c quita los comentarios al código que se proporciona para que puedas probar esta función.

```

/* Dados dos polinomios p1 y p2, devuelve 1 si todos los monomios del
polinomio p1 son también monomios del polinomio p2, y 0 en caso contrario.
Implementar este algoritmo en "Principal.c" y utilizando solo funciones de
"Polinomio.h"
*/
int estaIncluido(TPolinomio p1,TPolinomio p2);

```

Añade al fichero Polinomio.h el prototipo de la siguiente función, y añade su implementación al fichero Polinomio.c. En el fichero Principal.c quita los comentarios al código que se proporciona para que puedas probar esta función.

```
/* Evalúa el polinomio para el valor x y devuelve el resultado. Para la
evaluación del polinomio debes utilizar el método de Horner, de manera que
 $ax^4 + bx^3 + cx^2 + dx + e$  puede evaluarse en un valor cualquiera  $x$  teniendo en
cuenta que es equivalente a  $((ax + b)x + c)x + d)x + e$ .
*/
int evaluar(TPolinomio p,int x);
```

Anexo. Los prototipos de las funciones de lectura y escritura en ficheros de la biblioteca <stdio.h> son los siguientes (se dan por conocidos los prototipos de las funciones de <stdlib.h> que necesites, como free o malloc):

```
FILE *fopen(const char *path, const char *mode);
Abre el fichero especificado en el modo indicado ("rb"/ y "wb" para lectura/escritura binaria y "r"/"w"
para lectura/escritura de texto). Devuelve un puntero al manejador del fichero en caso de éxito y NULL
en caso de error.
```

```
int fclose(FILE *fp);
Guarda el contenido del buffer y cierra el fichero especificado. Devuelve 0 en caso de éxito y -1 en caso
de error.
```

```
unsigned fread(void *ptr, unsigned size, unsigned nmemb, FILE
*stream);
Lee nmemb elementos de datos, cada uno de tamaño size bytes, desde el fichero stream, y
los almacena en la dirección apuntada por ptr. Devuelve el número de elementos leídos.
```

```
int fscanf(FILE *stream, const char *format, ...)
```

Lee del fichero stream los datos con el formato especificado en el parámetro format, el resto de parámetros son las variables en las que se almacenan los datos leídos en el formato correspondiente. La función devuelve el número de variables que se han leído con éxito.