

Computación - Segundo Semestre 2010

Recuperatorio 18 de febrero de 2010

Importante: Crear un directorio de trabajo cuyo nombre sea *Nombre-Apellido*. Una vez finalizado el parcial ese directorio sólo debe contener los archivos fuente, y si se pidieron los script de gnuplot y los gráficos postscript. Todos los archivos deben contener en su nombre el apellido del autor.

Ejercicio 1: En el archivo *Estrellas.dat* aparecen listadas varias estrellas con sus correspondientes magnitudes aparentes y distancias a la Tierra (años luz). El archivo *Ccelestes.dat* contiene el nombre de varios objetos del Sistema Solar con sus respectivos radios (km) y albedos.

Hacer un programa que calcule las magnitudes absolutas de todos estos objetos. La magnitud absoluta de una estrella esta dada por:

$$M = m + 5 \log(d_0/d)$$

donde m es la magnitud aparente, y d es la distancia de la estrella a la Tierra y $d_0 = 10$ parsecs = 32616 años luz. Por otro lado, para los objetos del sistema solar la magnitud absoluta es:

$$H = m_S + 5 \log(d_0 \sqrt{a}/d_1)$$

donde $m_S = -26.73$ es la magnitud aparente del Sol a 1 UA, a es el albedo geométrico, d es el radio y $d_1 = 1 \text{ UA} = 1496108 \text{ km}$.

También debe determinarse la estrella y el cuerpo celeste de mayor brillo absoluto. (Recordar que un objeto es más brillante cuando más chico es el valor de su magnitud).

Armar una única subrutina de lectura que le pregunte al usuario el nombre del archivo a leer y que devuelva los datos obtenidos y el número de objetos vía common.

Las funciones que calculan las magnitudes absolutas deben ser *funciones de sentencia*.

Para calcular los objetos de mayor brillo utilizar un subprograma *function*. Escribir los resultados por pantalla con formatos y encabezados adecuados.

Ejercicio 2: Es posible representar un polinomio mediante un vector que contenga los coeficientes de éste. Así el polinomio $c_0 + c_1x + c_2x^2 + c_3x^3 + \dots + c_nx^n$ puede representarse como $(c_0, c_1, c_2, c_3, \dots, c_n)$. De igual forma la derivada de dicho polinomio $c_1 + 2c_2x + 3c_3x^2 + \dots + nc_nx^{n-1}$ puede representarse en forma de vector como $(c_1, 2c_2, 3c_3, \dots, nc_n)$

Escribir una subrutina, llamada *DERIVAPOL*, la cual recibirá un vector en su argumento representando un polinomio y arrojará como resultado otro vector representando la derivada de dicho polinomio, también vía argumento.

Escribir una función, llamada *EVALPOL*, para evaluar un polinomio. Dicha función recibirá como argumento un vector y una variable real. El vector contendrá los coeficientes de un polinomio y la variable real corresponderá al punto x donde se desea evaluar el polinomio. Esto nos permitirá evaluar tanto el polinomio como su derivada, dependiendo del vector que se le pase como argumento.

Escribir un programa que utilizando *DERIVAPOL* y *EVALPOL* calcule la derivada de un polinomio y evalúe el polinomio y su derivada en un dado punto. El punto donde evaluar, el grado del polinomio y vector que representa el polinomio deben ingresarse por teclado y los resultados escribirse por pantalla. El programa debe preguntarle al usuario si desea realizar un nuevo cálculo y en caso afirmativo pedir los nuevos datos.