

Computación - Primer Semestre 2009

Segundo Recuperatorio

2 de septiembre de 2009

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 *Acciones.dat* se da la cotización de las acciones en el mercado de valores argentino para el mediodía del 1 de septiembre de 2009. La primera columna se corresponde con el nombre de la acción, la segunda es el valor de la misma al mediodía (última cotización al momento de la consulta), la tercera al porcentaje de variación respecto del valor de cierre del día anterior, mientras que las dos últimas corresponden a los valores de apertura y cierre respectivamente.

Construir un programa que lea el archivo *Acciones.dat* y lo ordene de mayor a menor según la variación del valor de cada acción. Para ordenar debe usarse una subrutina y el resultado debe escribirse desde el programa principal con formato adecuado en un archivo cuyo nombre estará dado por el usuario. Por pantalla debe escribirse las acciones de mayor valor en el momento de la apertura, del cierre y de la última cotización. Para ello construir una *function* que para un dado vector devuelva el índice del elemento mayor.

NOTA: A priori no se sabe cuantas acciones hay en *Acciones.dat*.

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.