Prog. Modular e recursividade

Programación I

1º curso (1C)

Programación modular e recursividade

Unha vez que xa sabemos como implementar funcións que resolvan tarefas concretas, estamos en disposición de practicar a "programación modular". En varios exercicios pedirase que se implementen funcións recursivas que, como xa sabes, son as que se "invocan a si mesmas". Estas funcións conducen a implementacións moito mais elegantes e sinxelas.

Exercicio 1

Escribe unha función recursiva para calcular o enésimo termo da serie de Fibonacci. Emprega dita función para amosar en pantalla os N primeiros termos da serie, pedindo o valor de N ao usuario.

Exercicio 2

Escribe unha función que tome como argumento un número enteiro e devolva 1 se o número é primo, e 0 se non o é. Emprega esta función para obter o listado dos N primeiros números primos, sendo N un dato pedido ao usuario.

Exercicio 3

Escribe unha función recursiva que devolva o máximo común divisor de dous números. Para iso empregarás o algoritmo de Euclides, que funciona do seguinte xeito:

> *Dados dous números enteiros positivos m e n, tal que m > n,*
>
> *1) Dividimos m por n para obter o resto r*
>
> *2) Se r = 0, o MCD é n. Se non, o máximo común divisor é MCD(n,r).*

Escribe un programa que pida dous números ao usuario e calcule o seu MCD.

Exercicio 4

Escribe unha función recursiva que permita calcular x^y, sendo x e y números naturais, mediante multiplicacións.

Exercicio 5

Escribe unha función recursiva que permita sumar os díxitos dun número enteiro.

Exercicio 6

Escribe un programa que determine se dous números enteiros positivos calquera son “amigos”. Dous números son amigos se a suma dos divisores dun deles é igual ao outro número e viceversa. Por exemplo, os números 220 e 284 son amigos.

Posible mellora: escribe un programa que descubra todos os números amigos menores que 2000.