Ejercicios sobre funciones

Obligatorios

1. Los divisores propios de un entero *n* son sus divisores positivos menores que *n*. Un entero positivo se dice que es un número deficiente, perfecto, o abundante según si la suma de sus divisores propios es menor, igual o mayor que dicho número. Por ejemplo, 8 es deficiente porque 1+2+4=6<8, 6 es perfecto porque 1+2+3=6 y 12 es abundante porque 1+2+3+4+6=16>12. Escriba una función que reciba un entero positivo e y calcule la suma de sus divisores. A continuación, en el programa principal, utilice dicha función para averiguar si un número introducido por teclado es deficiente, perfecto o abundante.

2. Escriba una función que calcule el resultado de la siguiente integral como una suma discreta:

$$\int_0^L \frac{\sin^2(x)}{x^2}\,dx \approx \sum_{i=0}^{N} \frac{\sin^2(x_i)}{x_i^2}\Delta x$$

La función debe recibir, además de un real representando el límite superior *L*, un número entero *N* que represente el número de puntos de integración. Recuerde que $x_i = i\Delta x$, siendo la anchura de los subintervalos $\Delta x = L/N$. Compruebe que para *L* y *N* suficientemente grandes, el valor de la función será aproximadamente $\pi/2$, ya que

$$\int_0^\infty \frac{\sin^2(x)}{x^2}\,dx = \pi/2$$

Por ejemplo, para *L=200* y *N=1000000,* se obtiene 1,5682.

3. Decimos que un número entero es *guay* si puede obtenerse como suma de números enteros consecutivos; por ejemplo, 3 (=1+2), 5 (=2+3), 6 (=1+2+3), son números *guays*. Diseñe una función que reciba un número entero positivo e indique si éste es *guay*.

4. Recuerde que una función recursiva se define en términos de sí misma, y debe converger a un caso base (puede haber más de uno), definido completamente. A continuación se define el factorial de un número de forma recursiva:

$$n! = \begin{cases} 1 & si \quad n = 0 \\ n*(n-1)! & si \quad n > 0 \end{cases}$$

```
int factorial(int n)
{
	if(n==0) return 1;
	else return n*factorial(n-1);
}


int main(void)
{
	int p;
	....
	p=factorial(3);
	....
	return 0;
}
```

En este ejercicio se pide implementar una función recursiva que calcule el término enésimo de la sucesión de Fibonacci (1, 1, 2, 3, 5, 8, 13, 21...):

$$F(n) = \begin{cases} 1 \quad si \quad n = 0 \quad o \quad n = 1 \\ F(n-1) + F(n-2) \quad si \quad n > 1 \end{cases}$$

Una vez implementada la función *F(n)*, calcule el número áureo como:

$$\lim_{n\to\infty} \frac{F(n+1)}{F(n)}$$

y compruebe que su valor coincide aproximadamente con 1,61803398874989 para *n* suficientemente grande.

Voluntarios

5. Realice una función que indique si un número entero es primo.

6. Realice una función escriba un número entero descompuesto en factores primos.

Complementarios

7. ¿Qué devuelven las siguientes funciones recursivas?

a)
```
int f(int x)
{
     if(x>100) return x-10;
     else return f(f(x+11));
}
```

b)
```
int g(int n)
{
     if(n==0) return 0;
     else if(n%2==1) return g(n-1) + n;
     else return g(n-1)-n;

}
```