

Actividad 3.1

Algoritmos recursivos sencillos

Algoritmos y Estructuras de Datos

Tema 3: algoritmos recursivos

Escribe en C++, de forma recursiva, los siguientes procedimientos (no adjuntes un proyecto C++, sino que basta con poner en un Word el código de cada procedimiento, después de haberlo probado y asegurarte que funciona):

- El algoritmo de Euclides para calcular el MCD de dos números naturales. Los parámetros del método son los dos números. El resultado es su MCD. Precondiciones: los dos operandos son naturales y además el primero es mayor que el segundo.
- Un número real elevado a un número natural. Los parámetros del método son la base y el exponente. El resultado es la potencia. Precondiciones: el exponente es un número natural.
- La suma de los n primeros números naturales. El parámetro del método es n. El resultado es la suma. Precondición: el parámetro es un número natural.
- Comprobar si un número natural existe en un vector de números naturales. Si se encuentra se devuelve true y, en caso contrario, false. El parámetro es el vector (un puntero a su comienzo), su tamaño, y el número natural a encontrar dentro de él. El resultado es true o false. Precondiciones: el tamaño debe ser mayor o igual que 0 (si es 0, lógicamente se devolverá false).

Es necesario utilizar una nomenclatura adecuada. No es necesario poner comentarios ni precondiciones (aunque hay que tenerlas en cuenta a la hora de programar el código).

Solución

Euclides:

```
int calcularMCD (int operando1, int operando2) {  
    if (operando2==0)  
        return(operando1);  
    else  
        return (calcularMCD(operando2,(operando1%operando2)));  
}
```

Potenciación:

```
float elevarAPotencia (float base, int exponente) {  
    if (exponente == 0) return 1;  
    else return (base*elevarAPotencia(base,exponente-1));  
}
```

Sumatorio:

```
int sumarHasta (int fin) {  
    if (fin == 0) return 0;  
    else return (fin+sumarHasta(fin-1));  
}
```

Búsqueda:

```
bool buscar(int *vector, int n, int numeroABuscar) {  
    if (n==0) return false;  
    else  
        if (vector[n-1] == numeroABuscar) return true;  
    else return (buscar(vector, n-1, numeroABuscar));  
}
```