



EJERCICIOS

1. Implementa una función recursiva que calcule la suma de los n primeros números naturales. La cabecera sería la siguiente: unsigned sumanaturales(unsigned n) donde el parámetro n es el número de naturales a sumar.

```
#include <iostream>
using namespace std;

unsigned sumanaturales(unsigned n){
    unsigned suma;
    if(n == 0){
        suma = 0;
    }else{
        suma = n + sumanaturales(n-1);
    }
    return suma;
}

int main() {
    unsigned n;
    cout << "Suma de los n primeros números naturales." << endl;
    cout << "Introduce n: ";
    cin >> n;
    cout << "Suma = " << sumanaturales(n);
    return 0;
}
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



EJERCICIOS

1. El valor de la función potencia x^n , se puede definir recursivamente del modo siguiente:

$$x^n = 1 \quad \text{si } n=0 \quad x^n = x * x^{(n-1)} \quad \text{si } n \geq 1$$

Implementa una función potencia que calcule recursivamente el valor de x^n con la siguiente cabecera:

`unsigned potencia(unsigned x, unsigned n)`

```
#include <iostream>
using namespace std;
```

```
void leerDatos(unsigned& x, unsigned& n){
    cout << "Introduce x: ";
    cin >> x;
    cout << "Introduce n: ";
    cin >> n;
}
```

```
unsigned potencia(unsigned x, unsigned n){
    unsigned pot;
    if(n == 0){
        pot = 1;
    }else{
        pot = x * potencia(x, n-1);
    }
    return pot;
}
```

```
int main() {
    unsigned x, n;
    cout << "Programa que calcula una potencia x^n" << endl;
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



EJERCICIOS

1. Implementa una función recursiva que calcule el producto de dos números naturales x e y . La cabecera sería la siguiente:

```
unsigned producto(unsigned x, unsigned y)
```

A la hora de diseñar la solución ten en cuenta que los únicos operadores aritméticos que puedes usar son la suma y la resta.

```
#include <iostream>
using namespace std;
void leerDatos(unsigned& x, unsigned& y){
    cout << "Introduce x: ";
    cin >> x;
    cout << "Introduce y: ";
    cin >> y;
}
unsigned producto(unsigned x, unsigned y){
    unsigned prod;
    if(y == 0){
        prod = 0;
    }else{
        prod = x + producto(x, y-1);
    }
    return prod;
}
int main() {
    unsigned x, y;
    cout << "Producto de dos naturales. " << endl; // prints !!!Hello World!!!
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



EJERCICIOS

1. Implementa un procedimiento recursivo que imprima los dígitos de un número natural n en orden inverso. Por ejemplo, para $n=675$ la salida debería ser 576. La cabecera sería la siguiente: void inverso (unsigned n)

```
#include <iostream>
using namespace std;

unsigned leerDato(){
    unsigned res;
    cout << "Introduce número: ";
    cin >> res;
    return res;
}

void inverso (unsigned n){
    if(n == 0){
        cout << " ";
    }else{
        cout << n%10;
        inverso(unsigned(n/10));
    }
}

int main() {

    cout << "Invertir un número." << endl; // prints !!!Hello World!!!
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



EJERCICIOS

1. Implementa una función recursiva que devuelva true si el número que se le pasa como parámetro es primo y false en caso contrario. La cabecera de la función sería la siguiente:
bool esPrimo (unsigned num, unsigned divisor)
en el primer parámetro le habríamos de pasar el número, y el segundo parámetro es un número para que hay que comprobar si es o no divisor. Inicialmente (en la llamada a la función desde main) el valor de ese parámetro es 2.

```
#include <iostream>
using namespace std;
unsigned leerDato(){
    unsigned res;
    cout << "Introduce número: ";
    cin >> res;
    return res;
}
bool esPrimo (unsigned num, unsigned divisor){
    if(divisor < num){

        if(num%divisor != 0) {
            return esPrimo(num, divisor+1); // Si no encontramos un divisor, seguimos probando
        } else {
            return false; // Si encontramos un divisor de num, devolvemos false
        }
    }
    return true; //Devolvemos true si el divisor es mayor que el numero
}
int main() {
    unsigned n;
    bool b;
    cout << "Programa para saber si un número es o no primo." << endl; // prints !!!Hello World!!!
    n = leerDato();
    b= esPrimo(n, 2);
    if(b){
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99



EJERCICIOS

1. Implementa un procedimiento recursivo al que se le pase como parámetro un número n en base 10 (decimal), e imprima su valor en base 2 (binario). La cabecera sería la siguiente:

```
void decimalAbinario (unsigned n)
```

Por ejemplo para n=23, el procedimiento debería imprimir 10111.

```
#include <iostream>
using namespace std;
unsigned leerDato(){
    unsigned res;
    cout << "Introduce número: ";
    cin >> res;
    return res;
}
void decimalAbinario (unsigned n){
    if(n != 0){
        decimalAbinario(n/2);
        cout << n%2;
    }
}

int main() {
    unsigned n;
    cout << "Programa que pasa de decimal a binario." << endl; // prints !!!Hello World!!!
    n = leerDato();
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
CALL OR WHATSAPP:689 45 44 70

Cartagena99