

ej3.cpp.pdf



Anónimo



Fundamentos de la Programación



1º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga**



Creator Z16 HX Studio B13V
Tech meets Aesthetic

CAMBIO AQUÍ



MSI CONTENT CREATION
Creator Z16 HX Studio - B13V



¿Vas a acabar la carrera y todavía no se
te ha abierto el programa de edición?
Toca cambio.

Superportátil. Superpotencia para tus estudios.



Precios especiales
para
estudiantes
y profesores.

10%
dto
Mac

5%
dto
iPad



Escanea el QR
para saber más.

Rossellimac

Tu especialista
en Apple.

Descuento de 10% en Mac
y descuento de 5% en iPad
para estudiantes y profesores
mayores de 18 años. Oferta
válida hasta el 30/9/2023.

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

```
/*
```

3. Escribe un programa que acepte como entrada desde teclado un número natural mayor que cero

y dé como salida el resultado de sumar dos a dos los dígitos que aparecen en posiciones simétricas respecto al dígito central dentro del número dado como entrada. Por ejemplo :

para el número : 2354869

la salida es: $2+9 = 11$, $3+6 = 9$, $5+8 = 13$, 4

para el número : 6582

la salida es : $6+2 = 8$, $5+8 = 13$

```
*/
```

```
void leer_numero_mayor_0(int& numero){
    do{
        cout << "Introduzca un numero positivo: ";
        cin >> numero;
    }while(numero <= 0);
}
```

```
// Funcion auxiliar para calcular la cantidad de digitos de un numero
```

```
// Entrada: numero (int)
```

```
// Salida: int
```

```
int cantidad_digitos(int numero){
    int contador = 0;
    while(numero != 0){
        numero = numero/10;
        contador++;
    }
    return contador;
}
```

```
// funcion que obtiene el digito de un numero en una determinada posicion
```

```
// Entrada: n(int), i(int)
```

```
// Salida: int
```

```
int digito_n(int numero, int posicion){
    int resultado;
    if(posicion > cantidad_digitos(numero)){
        resultado = -1;
    }else{
        for(int i = 1; i <= posicion - 1; i++){
            numero = numero/10;
        }
    }
}
```

WUOLAH

```

    }
    resultado = numero % 10;
}
return resultado;
}

// Funcion que imprima la suma de los digitos simetricos
// Entrada: numero (int)
// Salida: NADA
void imprime_pares_sumitas(int numero){
    int cant_digitos = cantidad_digitos(numero);
    int x = cant_digitos;
    int y = 1;
    int d1, d2;
    for(int i = 1; i <= cant_digitos/2; i++){
        d1 = digito_n(numero, x);
        d2 = digito_n(numero, y);
        cout << d1 << " + " << d2 << " = " << d1 + d2 << endl;
        x--;
        y++;
    }
    numero = 12345;
    if(cant_digitos % 2 != 0){
        // Impar
        cout << digito_n(numero, (cant_digitos/2) + 1);
    }
}

int main(){
    int numero;
    leer_numero_mayor_0(numero);
    imprime_pares_sumitas(numero);

}

```