

Los números reales en coma flotante se convierten a binario en tres pasos:

1. Convertir al sistema binario
2. Escribir en notación científica
3. Seguir el standard IEEE754 para 32 bits

Por una parte la parte entera del número real se convierte a binario y por otra la parte decimal, según el algoritmo que se explica en el vídeo

<https://www.youtube.com/watch?v=VMcypTxcbvY>.

En esta práctica, se debe leer por teclado una base y un exponente máximo, debiéndose calcular la potencia para los posibles exponentes. Cada potencia, además, se debe convertir a binario.

Ej.: Con base=2 exponente=4 se obtendrá 1, 2, 4, 8, 16. Su conversión a binario sería: 1, 10, 100, 1000, 10000.

Ej.: Con base=10 exponente=3 se obtendrá 1, 10, 100, 1000. Su conversión a binario sería: 1, 1010, 1100100, 11 11101000.

ESPECIFICACIÓN: El cálculo de una potencia concreta se hará mediante un "for" (sin usar bibliotecas, ni funciones recursivas).

La base podrá estar entre 2 y 10, y el exponente entre 0 y 10. Aunque hayan sido introducidos correctamente, el usuario tendrá la opción de volver a introducir dichos datos.

ESPECIFICACIÓN: Estas comprobaciones se realizarán mediante "do while".

Cada grupo de 8 bits generados se separará por una línea en blanco, para tener mayor legibilidad.

ESPECIFICACIÓN: Para ello se utilizará el operador resto (módulo).

Cada vez que se terminen de printar en pantalla todos los bits de una conversión, se esperará a que pulsemos una tecla, para poder revisar los bits obtenidos, ya que el scroll automático de la pantalla nos haría perder los resultados.

ESPECIFICACIÓN: Se reutilizará el código de la práctica anterior (do while) y, como siempre, se debe seguir el algoritmo explicado en el vídeo.

```
base [2,10] ? 0
base [2,10] ? -1
base [2,10] ? 3
exponente [0,10] ? -1
exponente [0,10] ? 1
modificar base y exponente (S/N) ? s
```

```
base [2,10] ? 2
exponente [0,10] ? 4
modificar base y exponente (S/N) ? n
```

```
base: 2
exponente: 0
numero decimal= 1

bit0=1

PULSA PARA CONTINUAR
```

```
base: 2
exponente: 1
numero decimal= 2

bit0=0
bit1=1

PULSA PARA CONTINUAR
```

```
base: 2
exponente: 2
numero decimal= 4

bit0=0
bit1=0
bit2=1

PULSA PARA CONTINUAR
```

```
base: 2
exponente: 3
numero decimal= 8

bit0=0
bit1=0
bit2=0
bit3=1

PULSA PARA CONTINUAR
```

```
base: 2
exponente: 4
numero decimal= 16

bit0=0
bit1=0
bit2=0
bit3=0
bit4=1

PULSA PARA CONTINUAR
```

Figura 1. Ejemplo de ejecución del programa

```
/*
                                ESPERANZA MACARENA PLAZA MARTINEZ
                                Nº de matricula: br0427
*/
#include <stdio.h>
#include <stdlib.h>

int main () {
    int base, exponente, potencia, i, j, count = 0;
    char opcion = '\0', continuar;

    do {
        do {
            printf ("Escriba la base: ");
            fflush(stdin);
            scanf ("%d", &base);
        } while(base < 2 || base > 10);
        do {
            printf ("Escriba el exponente: ");
            fflush(stdin);
            scanf("%d", &exponente);
        } while(exponente < 0 || exponente > 10);
        printf ("Desea modificar la base y el exponente (S/N): ");
        fflush(stdin);
        scanf ("%c", &opcion);
    } while(opcion != 'N');

    //For para hacer las potencias
    for(i = 0; i <= exponente; i++) {
        potencia = 1;
        for (j = 1; j <= i; j++) {
            potencia *= base; //
        }
        printf ("\n");
        printf ("%d^%d = %d\n", base, i, potencia);

        //Pasar a binario
        do {
            printf("bit %d: %d\n", count, potencia % 2);
            potencia /= 2; // potencia = potencia / 2
            count++;
        } while (potencia > 0);
        printf("Desea coninuar? ");
        getchar();
    }
```

```
    fflush(stdin);  
    printf ("\n");  
    count = 0;  
}  
  
return 0;  
}
```