

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

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int bit, numbit, base, exp, decimal=1;
    float num, dec;
    char pregunta;

    do{
        do{
            printf("Base [2-10]: ");
            scanf("%i", &base);
        }while(base>10 || base<2);

        do{
            printf("Exponente [0-10]: ");
            scanf("%i", &exp);
            fflush(stdin);

        }while(exp>10 || exp<0);

        printf("Modificar base y exponente (S/N)? ");
        scanf("%c", &pregunta);
        fflush(stdin);

    }while(pregunta!='s' || pregunta!='S');

    for(int i=0; i<=exp; i++){
        printf("\nBase: %d\n", base);
        printf("Exponente: %d\n", i);
        if(i!=0)
            decimal*=base;
        printf("Decimal= %d\n", decimal);
        num=decimal;
        numbit=0;
        do{
            num/=2;
            dec=num-(int)num;
            if(dec >= 0.5)
                bit=1;
            else
                bit=0;
            if(numbit%8==0)
                printf("\n");
            printf("Bit%i=%i\n", numbit, bit);
            numbit++;
        }while(num>=1);

        printf("\n");
        system("pause");
    }
    return 0;
}
```