

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 fraccionaria, según el algoritmo que se explica en el vídeo

<https://www.youtube.com/watch?v=VMcypTxcbvY>. Este algoritmo deberá ser el utilizado, **no permitiéndose** el uso de otros algoritmos.

En esta práctica, apoyándonos en la **anterior**, y siguiendo las especificaciones de la **Figura 1**, se realizarán los siguientes pasos:

1. Convertir un número real con signo a **binario**
2. Escribir el número binario en **notación científica**
3. Representar el número binario según el **standard IEEE754** para 32 bits

```
numero real en base decimal ? -134.3125
numero real convertido a binario:
10000110.0101
numero binario en notacion cientifica:
1.00001100101
exponente: 7
exponente+127 en binario:
10000110
representacion en memoria:
11000011000000110010100000000000
```

Figura 1. Ejemplo de ejecución del programa

Se deben usar los prototipos, y defines, indicados en el siguiente recuadro:

```
// defines
#define maximo_chars 64

// prototipos usados en la práctica anterior
void convertir_entero(int, char [maximo_chars]);
// un int se convierte a binario (se almacena en el array)

void convertir_fraccionario(float , char [maximo_chars]);
// un float se convierte a binario (se almacena en el array)

void resetear(char [maximo_chars]); // se resetea el array

void colocar_posicion (char [maximo_chars],int , char );
// se coloca un char en la posicion int del array

int bits_blanco(char [maximo_chars]);
// chars en blanco en el array
```

```
void insertar_final(char [maximo_chars],char);
// se inserta un char al final del array, desplazando el resto a la izquierda

void printar_binario(char [maximo_chars]);
// se printa el array con los char del numero binario

// prototipos a definir en esta práctica

void mover_izda(char [maximo_chars]);
// mueve a la izda todos los bits del array

void scan_real(float * );
// scan del número real a convertir

int posicion_punto_decimal (char [maximo_chars]);
// posicion en el array de '.'

void notacion_cientifica(char [maximo_chars], int * );
// convierte el binario en notacion cientifica, transmitiendo el exponente

void copiar_mantisa(char [maximo_chars],char [32]);
// copia la mantisa a un array de 32 chars, en las últimos 23 posiciones del array

void copiar_exponente(char [maximo_chars],char r[32]);
// copia el exponente a un array de 32 chars, en las 8 siguientes posiciones a la posicion 0

void colocar_signo(char ,char [32]);
// coloca el signo en un array de 32 chars, en la posicion 0

int main(){
float real;
    int exponente;
    char cifra[maximo_chars], memoria[32]={ ' '}, s='0', exp[maximo_chars];
    resetear(cifra);
    resetear(exp);
    scan_real(&real);
    if(real<0)
        s='1';
    binario_entera(real, cifra);
    insertar_final(cifra, '.');
    binario_fraccionaria((real-(int)real), cifra);
    mover_izda(cifra);
    printf("Numero real convertido a binario: ");
    prn_binario(cifra);
    notacion_cientifica(cifra, &exponente);
    printf("\nNumero binario en notacion cientifica: ");
    prn_binario(cifra);
    printf("\nExponente: %i", exponente);
    binario_entera(exponente+127, exp);
    mover_izda(exp);
    printf("\nExponente+127 en binario: ");
    prn_binario(exp);
```

```
copiar_mantisa(cifra, memoria);
copiar_exponente(exp, memoria);
colocar_signo(s, memoria);
printf("\nRepresentacion en memoria: ");
for(int i=0; i<32; i++)
    printf("%c", memoria[i]);
return 0;}
```

```
// funciones a definir en esta práctica
```

```
void mover_izda(char numero[maximo_chars]){
    for(int i=bits_blanco(numero), j=0; i<maximo_chars; i++, j++){
        poner_posicion(numero, j, numero[i]);
        poner_posicion(numero, i, ' ');
    }
}

void scan_real(float *num){
    printf("Numero real en base decimal?: ");
    scanf("%f", num);
}

int posicion_punto_decimal(char numero[maximo_chars]){
    int i;
    for(i=0; numero[i]!='.'; i++){
    }
    return i;
}

void notacion_cientifica(char numero[maximo_chars], int *exp){
    int i,j;
    for(i=posicion_punto_decimal(numero)-1, j=0; i>0; i--, j++){
        poner_posicion(numero, i+1, numero[i]);
    }
    poner_posicion(numero, 1, '.');
    *exp=j;
}

void copiar_mantisa(char numero[maximo_chars], char mant[32]){
    int i,j;
    for(i=9, j=2; i<33; i++, j++){
        mant[i]=numero[j];
        if(mant[i]==' ')
            mant[i]='0';
    }
}

void copiar_exponente(char numero[maximo_chars], char r[32]){
    int i;
    for(i=0; i<8; i++){
        r[i+1]=numero[i];
        if(r[i]==' ')
            r[i]='0';
    }
}

void colocar_signo(char signo, char r[32]){
    r[0]=signo;
}
```