

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 se hará lo mismo que en la práctica 3, pero definiendo y usando funciones y arrays. La conversión a binario se hará usando un array con un máximo de chars definido en la parte #define del programa.

```
entera ? 134
fraccionaria ? .3125
10000110.0101
Process returned 0 (0x0)   execution time : 7.575 s
Press any key to continue.
```

Figura 1. Ejemplo de ejecución del programa

Se deben usar los prototipos indicados en el siguiente recuadro.

```
// ESPERANZA MACARENA PLAZA MARTINEZ
// N° de matricula: br0427

#include <stdio.h>
#include <stdlib.h>
#define maximoChars 64

//Prototipos
void binarioEntera(int, char []);
// un int se convierte a binario (se almacena en el array)

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

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

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

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

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

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

int main() {
    int entero = 0;
    float decimal = 0.0;
    char binary[maximoChars] = {0};

    resetear(binary);
    printf("Escriba la parte entera de un numero: ");
    fflush(stdin);
    scanf("%d", &entero);
    binarioEntera(entero, binary);
    printf("Escriba la parte decimal de un numero: ");
    fflush(stdin);
    scanf("%f", &decimal);
    binarioFraccionaria(decimal, binary);
    printBinario(binary);

    return 0;
}

void binarioEntera(int num, char binary[maximoChars]){
    int i = maximoChars - 1;

    while(num > 0){
        if(10 * (((float) num / 2) - (num / 2)) >= 5){
            binary[i--] = '1';
        } else{
            binary[i--] = '0';
        }
        num /= 2;
    }
}

//Definicion de funciones
```

```
void binarioFraccionaria(float decimal, char binary[maximoChars]){
    insertarFinal(binary, '.');
    while(decimal != 0.0){
        if((int) (decimal * 2) != 0){
            insertarFinal(binary, '1');
        } else{
            insertarFinal(binary, '0');
        }
        decimal = decimal * 2 - ((int) (decimal * 2));
    }
}

void resetear(char binary[maximoChars]){
    for(int i = 0; i < maximoChars; i++){
        binary[i] = ' ';
    }
}

void ponerPosicion (char binary[maximoChars],int pos, char car){
    binary[pos] = car;
}

int bitsBlanco(char cad[maximoChars]){
    int count = 0;

    for(int i = 0; i < maximoChars; i++){
        if(cad[i] == ' '){
            count++;
        }
    }
    return count;
}

void insertarFinal(char cad[maximoChars], char car){
    for(int i = 0; i < maximoChars; i++){
        cad[i] = cad[i + 1];
    }
    cad[maximoChars - 1] = car;
}

void printBinario(char binary[maximoChars]){
```

```
    for(int i = 0; i <= maximoChars; i++){  
        if(binary[i] != ' '){  
            printf("%c", binary[i]);  
        }  
    }  
}
```