

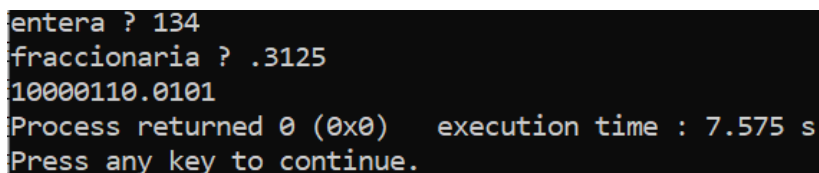
Los números reales en coma flotante se convierten a binario en tres pasos:

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.

```
// defines
#include <stdio.h>
#include <stdlib.h>
#define maximo_chars 64

// prototipos
void binario_entera(int, char [maximo_chars]);
// un int se convierte a binario (se almacena en el array)

void binario_fraccionaria(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 poner_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 c);
// se inserta un char al final del array, desplazando el resto a la izquierda
```

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

int main(){
    float num, num2;
    char cifra[maximo_chars];
    resetear(cifra);
    do{
        printf("Entera?: ");
        scanf("%f", &num);
    }while(num<0);

    do{
        printf("Fraccionaria?: ");
        scanf("%f", &num2);
    }while(num2<0 || num2>=1);

    binario_entera(num, cifra);
    if(num2!=0)
        insertar_final(cifra, '.');
    binario_fraccionaria(num2, cifra);
    prn_binario(cifra);
    return 0;
}

// definicion de las funciones
void binario_entera(int entera, char numero[maximo_chars]){
    float dec, ent=entera;
    for(int i=maximo_chars-1; ent>=1; i--){
        ent/=2;
        dec=ent-(int)ent;
        if(dec>=0.5)
            numero[i]='1';
        else
            numero[i]='0';
    }
}

void binario_fraccionaria(float decimal, char numero[maximo_chars]){
    do{
        decimal*=2;
        if(decimal<1)
            insertar_final(numero, '0');
        else{
            insertar_final(numero, '1');
            decimal-=1;
        }
    }while(decimal!=0);
}
```

```
void resetear(char numero[maximo_chars]){
    for(int i=0;i<maximo_chars;i++)
        numero[i]=' ';
}

void poner_posicion (char numero[maximo_chars],int n, char caracter){
    numero[n]=caracter;
}

int bits_blanco(char numero[maximo_chars]){
    int cont=0;
    for(int i=0; i<maximo_chars; i++)
        if(numero[i]==' ')
            cont++;
    return cont;
}

void insertar_final(char numero[maximo_chars],char c){
    for(int i=bits_blanco(numero)-1; i<maximo_chars-1; i++)
        poner_posicion(numero, i, numero[i+1]);
    poner_posicion(numero, maximo_chars-1, c);
}

void prn_binario(char numero[maximo_chars]){
    for(int i=0; i<maximo_chars; i++)
        if(numero[i]!=' ')
            printf("%c", numero[i]);
}
```