Operaciones con números en formato punto flotante

Ejercicio 1
Compile el siguiente código en C en su PC:

```
#include <stdio.h>
#include <float.h>
#include <math.h>
#include <fenv.h>

int main(void)
{
    float a, b, c, f1, f2;
    double d1;

    a = 1000000000.0;
    b =   20000000.0;
    c =   20000000.0;

    f1 = (a * b) * c;
    f2 = a * (b * c);

    d1 = (double) (a) * (double) (b) * (double) (c);

    printf("f1 = %f \n", f1 );
    printf("f2 = %f \n", f2 );
    printf("d1 = %lf \n", d1 );

    printf("Error en f1 = %10e \n", f1 - 4000000000000000000000000.0 );
    printf("Error en f2 = %10e \n", f2 - 4000000000000000000000000.0 );
    printf("Error en d1 = %20e \n", d1 - 4000000000000000000000000.0 );

    double acum_1, acum_2;

    acum_1 = 0.0;
    for (int i = 0; i < 10000000; i++){ acum_1 += 0.01; }

    acum_2 = 0.0;
    b = 0.333;
    for (int i = 0; i < 10000000; i++){ acum_2 += b / b; }

    printf("acum_1 = %.20lf \n", acum_1 );
    printf("acum_2 = %.20lf \n", acum_2 );

    printf("Error en acum_1 = %.20lf \n", acum_1 - (100000));
    printf("Error en acum_2 = %.20lf \n", acum_2 - (10000000));

    return 0;
}
```

1. Inspeccione el código y determine el objetivo del programa
2. Analice los valores de las variables `f1`, `f2`, `d1`, `acum_1` y `acum_2`.
3. Cambie el tipo de dato de `acum_1` y `acum_2` a `float`. Compile y vuelva a ejecutar. Analice el nuevo valor de los errores.
4. ¿Qué conclusión puede obtener a partir de estos valores?

Ejercicio 2. Modos de redondeo.
Analice y compile el archivo `ex_02.c`.

1. ¿Para qué sirven las funciones `fegetround()` y `fesetround()`?
2. ¿Cuál es el modo de redondeo por defecto con el que arranca el programa?
3. Ejecute el programa para los modos de redondeo FE_DOWNWARD, FE_UPWARD y FE_TOWARDZERO, y compárelos con el modo FE_TONEAREST.
4. ¿Observa diferencias? ¿Estas diferencias son consistentes con los modos de redondeos?

Ejercicio 3. Valores especiales.
Investigue como inicializar en C variables en punto flotante con los valores especiales NaN, Inf e -Inf. Además, escriba un programa en C que produzca como resultado estos valores especiales.

Ejercicio 4. Excepciones.
Analice y compile el archivo `ex_04.c`.

1. Analice los resultados impresos por consola ¿Son los resultados consistentes con las operaciones ejecutadas?
2. Explique qué hacen las funciones `feclearexcept()`, `feraiseexcept()` y `fetestexcept()`.

Ejercicio 5. Manejo de excepciones.
Analice y compile el archivo `ex_05.c`.

3. Analice los resultados impresos por consola ¿Son los resultados consistentes con las operaciones ejecutadas?
4. Descomente las líneas 38 a 43 y vuelva a compilar.
5. ¿Qué observa por consola? ¿Cuál es la función de `feenableexcept()`?
6. Descomente la línea 36 y vuelva a compilar.
7. Qué observa por consola? ¿Cuál es la función de `signal(SIGFPE, fpe_handler)`?

Ejercicio 6. Excepciones.
Utilizando el código visto en el ejercicio 4, `ex_04.c`. genere los 4 tipo de excepciones estudiadas pero ejecutando diferentes operaciones matemáticas en lugar de utilizar la función `feraiseexcept()`.