$f(x)=x^3-2x+2$ と定義する。Newton 法で方程式 $f(x)=0$ の実数解の近似値を計算することを考える。Newton 法とは、与えられた初期値 x_0 に対して、次の漸化式により反復計算を行う手法である。

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)} \quad (n \geq 0) \tag{1}$$

方程式 $f(x)=0$ はただ一つの実数解

$$a=\sqrt[3]{-1+\sqrt{\frac{19}{27}}}+\sqrt[3]{-1-\sqrt{\frac{19}{27}}} \tag{2}$$

を持つことは既知として、以下の問いに答えよ。

1. 漸化式 (1) を用いて、$x_1, x_2, \ldots$ を反復的に計算する関数 `newton(x0)` を, Julia 言語で次の要件を満たすように作成せよ。
   (a) 引数 `x0` を初期値とすること
   (b) $f'(x_n)=0$ のとき、反復計算を終了する
   (c) 各 $n \geq 0$ に対して、[x_n **の値**] と [**誤差** $|x_n-a|$] を表示する
   (d) x_{n+1} を計算した時点で、$|x_{n+1}-x_n|<10^{-6}$ が満たされているならば、反復計算を終了する

   各要件がプログラムコードのどこの部分に該当するのかを明記し、プログラムに関する説明を必ず記述すること。

```
function newton(x)
    # 引数 x を初期値とする ## 1 ##
    i=0
    while true
        println(i, "回目")
        # 計算の都度 近似解とその誤差を表示 ## 3 ##
        println("+-- x の値")
        println(x)
        println("+-- aとの誤差")
        println(abs(x-a))
        # 分母の微分係数が 0 になれば終了 ## 2 ##
        if f2(x) == 0
```

```
            println("f'(x)=0 により終了")
            return 0
        end
        # 漸化式の計算
        y = x - f1(big(x))/f2(big(x))
        # 近似解の差が小さい場合終了 ## 4 ##
        if abs(y-x) < big(10)^(-6)
            println("近似解の差が小さいので終了")
            return 0
        end
        # 30回を超えたら中止
        if i > 30
            println("30回を超えたので中止")
            return 1
        end
        x=y # 近似解の項を次にセットする
        i += 1
    end
end
```

---

2. `newton(-16)` および `newton(-2)` の実行結果を示し、誤差や収束の様子について考察を述べよ。

 ....

 方程式の実数解は $a = -1.769292\cdots$ であるので、初期値はこの値に近いと早くアルゴリズムが終わる。`newton(-16)` は x_9 まで計算し、`newton(-2)` は x_3 までの計算で近似解が得られた。

---

3. 初期値 `x0` を 0 に近い値に選び、`newton(x0)` の実行結果を示せ。さらに、実行結果に関する考察を述べよ。

 ....

 漸化式 $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ は $x_n = 1$ のとき、$x_{n+1} = 0$ となり、$x_n = 0$ のとき、$x_{n+1} + 1$ となる。

 この為、数列 $\{x_i\}_{i\in\mathbb{N}}$ はどこかで 0 か 1 が現れると、それ以降 $0, 1$ が交代に出て

くるだけになる。近似解に近づかない数列となるのでこれを避けるように初期値を選択しないといけない。