```
Input: function f(x), float x_0, float x_1, float x_2, int iterations
Output: solution vector results

begin muller

    if (tolerance or iterations or x_0 or x_1 or x_2 is not a valid numbers):
        break;
    if (x_1 is equals to x_2):
        break;
    array results
    if ((function(x_0) > 0 and function(x_1) < 0) or (function(x_0) < 0
    and function(x_1) > 0)) then
        int count <- 0
        float error <- |x_1 - x_2|
        while ((error > tolerance) and (count < iterations)) do
            float h_0 <- x_1 - x_0
            float h_1 <- x_2 - x_1
            float f_x0 <- function(x_0)
            float f_x1 <- function(x_1)
            float f_x2 <- function(x_2)
            float delta_0 <- (f_x1 - f_x0) / h_0
            float delta_1 <- (f_x2 - f_x1) / h_1
            float a <- (delta_1 - delta_0) / (h_1 - h_0)
            float b <- (a * h_1) + delta_1
            float c <- f_x2
                float aux <- (b ** 2) - (4 * a * c)
                if (aux < 0) then
                        break;
            float raiz <- raiz((b ** 2) - (4 * a * c))
            if (b < 0) then
                denominador = b - raiz
            else:
                denominador = b + raiz
            x_3 <- x_2 + ((-2*c)/denominador)
            x_0 <- x_1
            x_1 <- x_2
            x_2 <- x_3
            error <- |x_1 - x_2|
                array iteration <- [count,x_2,f_x2,error]
            results[count] <- iteration
            count <- count + 1
        end while
    return results

end muller
```