

Practical No: 02

Aim: Implement Fibonacci Search

Code:

Step 1: Define the Function

```
function f(x)
    return x * x - x + 1
end
```

```
julia> function f(x)
    return x*x-x+1
end
f (generic function with 1 method)

julia>
```

Step 2: Write a function of Fibonacci Search

```
function fibonacci_search(f, a, b, n; ε = 0.01)
    s = (1 - √5) / (1 + √5)
    ρ = 1 / (1.618 * (1 - s^(n + 1)) / (1 - s^n))
    d = ρ * b + (1 - ρ) * a
    yd = f(d)
    for i in 1:n-1
        print(a)
        print("\n")
        print(b)
        print("\n")
        if i == n - 1
            c = ε * a + (1 - ε) * d
        else
            c = ρ * a + (1 - ρ) * b
        end
        yc = f(c)
        if yc < yd
            b, d, yd = d, c, yc
        else
            a, b = b, c
        end
        ρ = 1 / (1.618 * (1 - s^(n - i + 1)) / (1 - s^(n - i)))
    end
end
```

```

end
return a < b ? (a, b) : (b, a)
end

```

Step 3: Call Fibonacci Search with the defined function and an interval (a, b) and number of iterations

In our case $a = 2$ and $b = 4$ and number of iterations $n = 10$

```

julia> fibonacci_search(f,2,4,10)
2
4
2
3.235981019447566
2
2.764018980552434
y 2
r 2.471903975250067
2
2.2921149914180354
2
2.179766806259804
2
2.1123481435573277
2
2.0674101921616197
2
2.0449378413884616
(2, 2.0224691200126035)

```