

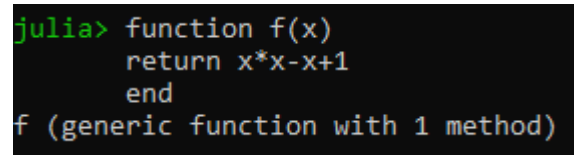
Algorithm for Optimization

Practical No.3

AIM: Implement Golden section search.

1. Code for Function Creation:

```
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)
```

2. Code for Algorithm:

```
function golden_section_search(f, a, b, n)
    ρ = 1.618-1
    d = ρ * b + (1 - ρ)*a
    yd = f(d)
    for i = 1 : n-1
        print(a,"\n")
        print(b,"\n")
        c = ρ*a + (1 - ρ)*b
        yc = f(c)
        if yc < yd
            b, d, yd = d, c, yc
        else
            a, b = b, c
        end
    end
    return a < b ? (a, b) : (b, a)
end
```

```

julia> function golden_section_search(f, a, b, n)
    ĩ = 1.618-1
    d = ĩ * b + (1 - ĩ)*a
    yd = f(d)
    for i = 1 : n-1
        print(a, "\n")
        print(b, "\n")
        c = ĩ*a + (1 - ĩ)*b
        yc = f(c)
        if yc < yd
            b, d, yd = d, c, yc
        else
            a, b = b, c
        end
    end
    return a < b ? (a, b) : (b, a)
end
golden_section_search (generic function with 1 method)

```

3. Output For Code:

golden_section_search(f,2,3,5)

```

julia> golden_section_search(f,2,3,5)
2
3
2
2.6180000000000003
2
2.3819999999999997
2
2.2360759999999997
(2, 2.145924)

```