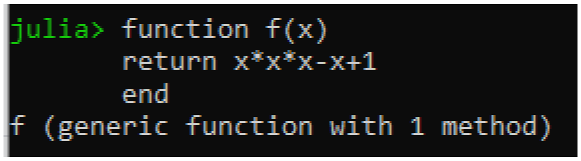
Practical No. 04

Aim : Implement Quadratic Fit Search.

Step 1: First Define Function

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



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

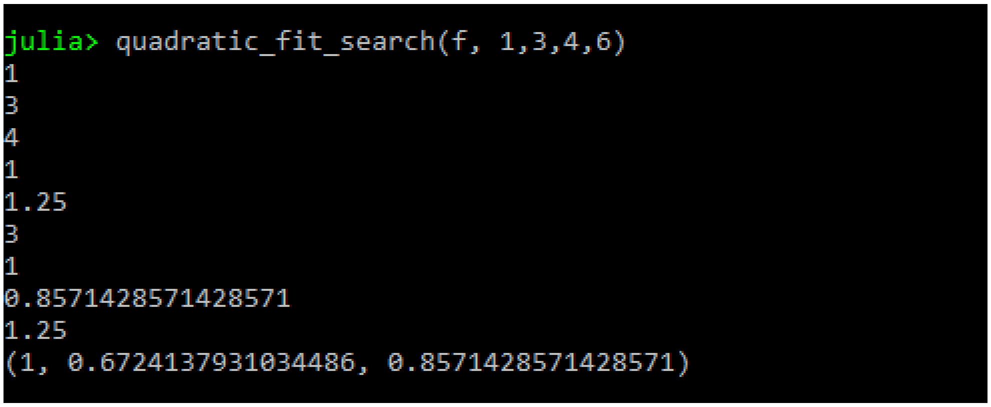
Step 2: Quadratic Fit Algorithm

```
function quadratic_fit_search(f, a, b, c, n)
    ya, yb, yc = f(a), f(b), f(c)
    for i in 1:n-3
        print(a,"\n",b,"\n",c,"\n")
        x = 0.5*(ya*(b^2-c^2)+yb*(c^2-a^2)+yc*(a^2-b^2)) /(ya*(b-c)
+yb*(c-a) +yc*(a-b))
        yx = f(x)
        if x > b
           if yx > yb
              c, yc = x, yx
            else
              a, ya, b, yb = b, yb, x, yx
        end
        else if x < b
                if yx > yb
                   a, ya = x, yx
                else
                   c, yc, b, yb = b, yb, x, yx
                end
            end
        end
    return (a, b, c)
end
```

Step 3: For Output, Call the function.

quadratic_fit_search(f, 1, 3, 4, 6)

Output :



```
julia> quadratic_fit_search(f, 1,3,4,6)
1
3
4
1
1.25
3
1
0.8571428571428571
1.25
(1, 0.6724137931034486, 0.8571428571428571)
```