

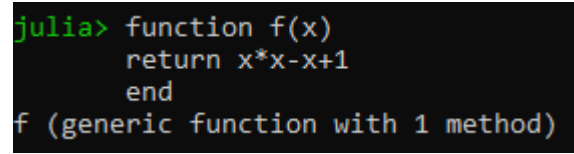
Algorithm for Optimization

Practical No.4

AIM: Implement Quadratic Fit 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 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
elseif 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

```

```

julia> 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
        elseif x < b
            if yx > yb
                a, ya = x, yx
            else
                c, yc, b, yb = b, yb, x, yx
            end
        end
        end
        end
        return (a, b, c)
    end
quadratic_fit_search (generic function with 1 method)

```

3. Output For Code:

quadratic_fit_search(f,2,3,4,5)

```

julia> quadratic_fit_search(f,2,3,4,5)
2
3
4
2
0.5
3
(2, 0.5, 3)

```