

Project 1 - MATH2504 - 2025

Author: Benjamin Nicholson, s4697411

[Open GitHub Repo](#)

This project is largely incomplete, makes for easy marking

```
In [1]: #example script task 1

using Pkg
Pkg.activate(".")
include("poly_factorization_project.jl")

#initialise polynomials
x = x_poly(PolynomialDense{Int,Int})
f = 2x^3 + 4x^2 - 3x
g = 2x^4 - 4x^2 - 3x + 3
h = 3x^2 + 13x^4 - 8x

println("Task 1.2: Compute f + g, f * h, & g * h")
@show f+g
@show f*h
@show g*h

println("Task 1.3: Compute derivative(f * g) and show it is equal to what is expected")
@show derivative(f*g)
@show derivative(f)*g + f*derivative(g)

println("Task 1.4: compute (f * h) ÷ h modulo 5, 17 and 101, and confirm using mod")
mod5q, mod5r = div_rem_mod_p(f*h,h,5)
mod17q, mod17r = div_rem_mod_p(f*h,h,17)
mod101q, mod101r = div_rem_mod_p(f*h,h,101)
fmod5, fmod17, fmod101 = mod(f,5), mod(f,17), mod(f,101)

println("f * h ÷ h mod 5 = ", mod5q)
println("fmod5 = ", fmod5)
@show fmod5 == mod5q

println("f * h ÷ h mod 17 = ", mod17q)
println("fmod17 = ", fmod17)
println("fmod101 = ", fmod101)

println("f * h ÷ h mod 101 = ", mod101q)
println("fmod101 = ", fmod101)

println("Task 1.5: compute gcd_mod_p(f*h, g*h, p) for p equal to 5, 11 and 13.")
p = [5, 11, 13]
for i in p
    println("for p = ", i, ", gcd_mod_p(f*h, g*h, ",i,") = ", gcd_mod_p(f*h, g*h, i
```

```
end
```

Activating project at `C:\Users\Studying\Desktop\MATH2504\Project 1 v2\Benjamin-Nicholson-2504-2025-PROJECT1`

Task 1.2: Compute $f + g$, $f * h$, & $g * h$

$$f + g = 2x^4 + 2x^3 - 6x + 3$$

$$f * h = 26x^7 + 52x^6 - 33x^5 - 4x^4 - 41x^3 + 24x^2$$

$$g * h = 26x^8 - 46x^6 - 55x^5 + 27x^4 + 23x^3 + 33x^2 - 24x$$

Task 1.3: Compute derivative($f * g$) and show it is equal to what is expected when utilizing product rule

$$\text{derivative}(f * g) = 28x^6 + 48x^5 - 70x^4 - 88x^3 + 18x^2 + 42x - 9$$

$$\text{derivative}(f) * g + f * \text{derivative}(g) = 28x^6 + 48x^5 - 70x^4 - 88x^3 + 18x^2 + 42x - 9$$

Task 1.4: compute $(f * h) \div h$ modulo 5, 17 and 101, and confirm using mod (in polynomial_dense.jl) that f modulo 5, 17, and 101 is returned.

$$f * h \div h \bmod 5 = 2x^3 + 4x^2 + 2x$$

$$f \bmod 5 = 2x^3 + 4x^2 + 2x$$

$$f \bmod 5 == \text{mod5} = \text{true}$$

$$f * h \div h \bmod 17 = 2x^3 + 4x^2 + 14x$$

$$f \bmod 17 = 2x^3 + 4x^2 + 14x$$

$$f \bmod 101 = 2x^3 + 4x^2 + 98x$$

$$f * h \div h \bmod 101 = 2x^3 + 4x^2 + 98x$$

$$f \bmod 101 = 2x^3 + 4x^2 + 98x$$

Task 1.5: compute $\text{gcd_mod_p}(f*h, g*h, p)$ for p equal to 5, 11 and 13.

$$\text{for } p = 5, \text{gcd_mod_p}(f*h, g*h, 5) = 4x^4 + 4x^2 + x$$

$$\text{for } p = 11, \text{gcd_mod_p}(f*h, g*h, 11) = 5x^4 + 2x^2 + 2x$$

$$\text{for } p = 13, \text{gcd_mod_p}(f*h, g*h, 13) = 9x^3 + 10x^2 + 9x$$

In [2]: *#example script task 2*

```
using Pkg
Pkg.activate(".")

include("poly_factorization_project.jl")

x1 = x_poly(PolynomialDense{BigInt,Int})
x2 = x_poly(PolynomialDense{Int,Int})

p1intint = (10*10^18)*x2^2
p2intint = (10*10^17)*x2^2

p1bigintint = big(10)*10^18*x1^2
p2bigintint = big(10)*10^17*x1^2

@show p2intint+p1intint
@show p1bigintint+p2bigintint
```

Activating project at `C:\Users\Studying\Desktop\MATH2504\Project 1 v2\Benjamin-Nicholson-2504-2025-PROJECT1`

$$p2intint + p1intint = -7446744073709551616x^2$$

$$p1bigintint + p2bigintint = 11000000000000000000x^2$$

Out[2]: 11000000000000000000x^2

Task 2.9 I am unable to run the tests in the Jupyter notebook, this is the output from VSCode

```

Activating project at C:\Users\Studying\Desktop\MATH2504\Project 1
v2\Benjamin-Nicholson-2504-2025-PROJECT1 ---BEGIN INTEGER UNIT TESTS---
test_euclid_ints - PASSED test_ext_euclid_ints - PASSED ---END INTEGER UNIT TESTS---

---BEGIN HEAP UNIT TESTS--- test_heap - popall non-destructive vector constructor -
PASSED test_heap - PASSED ---END HEAP UNIT TESTS---

---BEGIN POLYNOMIAL UNIT TESTS---

Int, Int64} - PASSED prod_derivative_test_poly for PolynomialDense{BigInt, Int64} - PASSED
ext_euclid_test_poly for PolynomialDense{BigInt, Int64} - PASSED division_test_poly for
PolynomialDense{BigInt, Int64} - PASSED

---END POLYNOMIAL UNIT TESTS---

---BEGIN FACTORIZATION UNIT TESTS---

Type of `Polynomial`: PolynomialDense{Int64, Int64}

doing prime = 5 ..... doing prime = 7 ..... doing prime = 11 .....

factor_mod_p_test_poly for PolynomialDense{Int64, Int64} - PASSED

Type of `Polynomial`: PolynomialDense{BigInt, Int64}

doing prime = 5 ..... doing prime = 7 ..... doing prime = 11 .....

factor_mod_p_test_poly for PolynomialDense{BigInt, Int64} - PASSED

---END FACTORISATION UNIT TESTS---

```

In [3]:

```

UndefVarError: `jupyter` not defined in `Main`
Suggestion: check for spelling errors or missing imports.

```

Stacktrace:

```

[1] top-level scope
      @ In[3]:1

```

In []: