

<https://github.com/georgerapeanu/BBU-Computer-Science/tree/master/Semester5/Formal Languages and Compiler Design/lab3>

p1.out

Program: let x: i32;let y: i32;x = readI32();y = readI32();while y != 0 {let z: i32 = x % y;

PIF.out

let: -

x: 0

:: -

i32: -

:: -

let: -

y: 1

:: -

i32: -

:: -

x: 0

=: -

readI32: -

(: -

): -

:: -

y: 1

=: -

readI32: -

(: -

): -

:: -

while: -

y: 1

!: -

=: -

0: 2

{: -

let: -

z: 3

:: -

i32: -

=: -

x: 0

?: -

y: 1

:: -

x: 0

=: -

y: 1

```

;; -
y: 1
=: -
z: 3
;; -
}: -
print: -
(: -
"Gcd is ": 4
,: -
x: 0
): -
;; -
ST.out
x: 0
y: 1
0: 2
"Gcd is ": 4
z: 3

```

p2.out

```

Program: let n: i32 = readI32();let i: i32 = 0;let is_prime: bool = true;i = 2;while i < n + 1 do
PIF.out
let: -
n: 0
:: -
i32: -
=: -
readI32: -
(: -
): -
;; -
let: -
i: 1
:: -
i32: -
=: -
0: 2
;; -
let: -
is_prime: 3
:: -
bool: -
=: -
true: 4
;; -

```

```

i: 1
=: -
2: 5
;: -
while: 6
i: 1
<: -
n: 0
{: -
if: -
n: 0
%: -
i: 1
==: -
0: 2
{: -
is_prime: 3
=: -
false: 7
;: -
}: -
i: 1
=: -
i: 1
+: -
1: 8
;: -
}: -
if: -
is_prime: 3
{: -
print: -
(: -
"Number is prime": 9
): -
;: -
}: -
else: -
{: -
print: -
(: -
"Number is not prime": 10
): -
;: -
}: -
ST.out

```

```

0: 2
1: 8
2: 5
false: 7
"Number is prime": 9
i: 1
is_prime: 3
while: 6
"Number is not prime": 10
n: 0
true: 4

```

```
p3.out
```

```

Program: let n: u32;let sum: u32 = 0;let i: u32 = 0;n = readU32();while i < n {let val: u32
PIF.out
let: -
n: 0
:: -
u32: -
;: -
let: -
sum: 1
:: -
u32: -
=: -
0: 2
;: -
let: -
i: 3
:: -
u32: -
=: -
0: 2
;: -
n: 0
=: -
readU32: -
(: -
): -
;: -
while: -
i: 3
<: -
n: 0
{: -
let: -

```

```

val: 4
:: -
u32: -
=: -
readU32: -
(: -
): -
;: -
sum: 1
=: -
sum: 1
+: -
val: 4
;: -
i: 3
=: -
i: 3
+: -
1: 5
;: -
}: -
print: -
(: -
"Sum is": 6
,: -
sum: 1
): -
;: -
ST.out
0: 2
i: 3
"Sum is": 6
sum: 1
1: 5
n: 0
val: 4

plerr.out

Lexical error for token number 2 "0a: i32; let x = 12_34; "
note: run with 'RUST_BACKTRACE=1' environment variable to display a backtrace

```