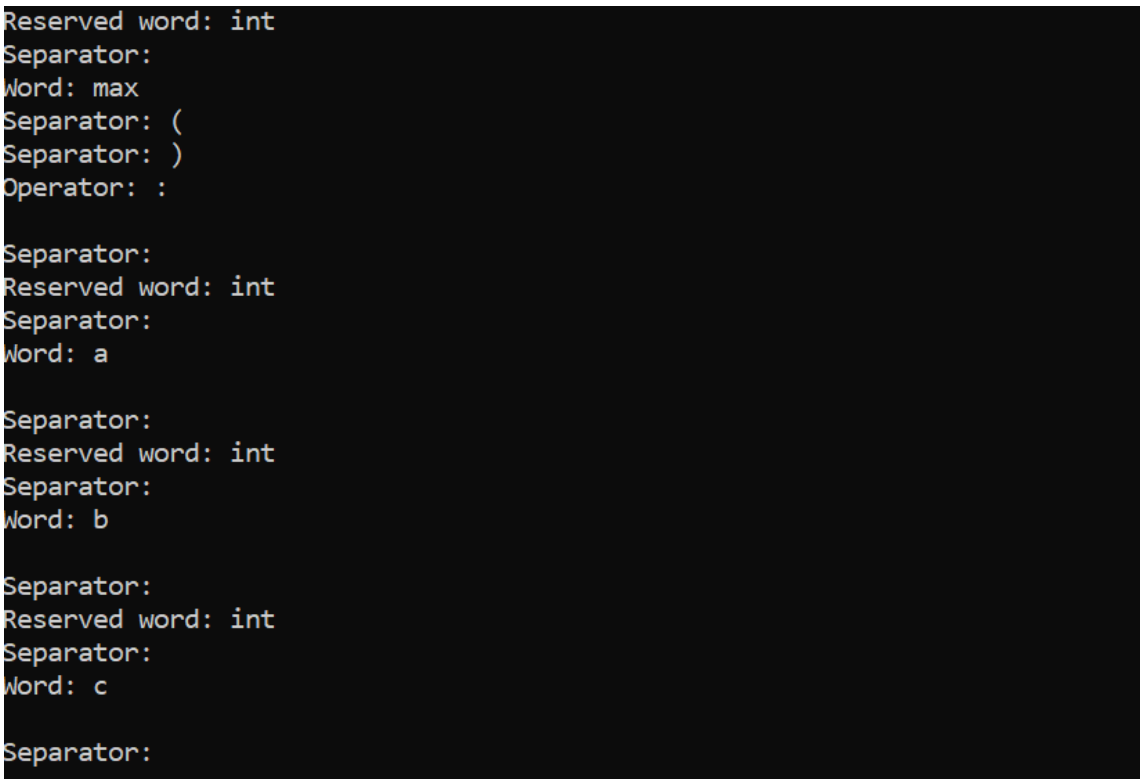
Github link: https://github.com/pelotazos123/FLCD-erasmus/tree/main/Lab8

Lexical analyzer implemented in C using FLEX



```
C:\Users\pablo\Documents\LEXFLCD>test1.exe p2.txt
Reserved word: int
Separator:
Word: gcd
Separator: (
Separator: )
Operator: :

Separator:
Reserved word: int
Separator:
Word: a

Separator:
Reserved word: int
Separator:
Word: b

Separator:
Reserved word: int
Separator:
Word: x
Separator:
Set: =
Separator:
Char: 1
```

P1 used:

```
int max():
        int a
        int b
        int c


        if (a > b)
                if (a > c)
                        return a
                return c
        if (b > c)
                return b
        return c


int min():
```

```
        int a
        int b
        int c

        if (a < b)
                if (a < c)
                        return a
                return c

        if (b < c)
                return b
        return c

bool prime(int n):
        for (int i = n; i >= 0; i--)
                if (n%i==0)
                        return false
        return true
```

P2 used:

```
int gcd():
        int a
        int b
        int x = 1

        for (int l = 1; l <= a & l <= b; i++)
                if (a%i == 0 & n2 % i == 0)
                        x = i
        return x

int equation():
        int a
```

```
int b

int c

int x1

int x2

x1 = (-b + (b^2 – 4 * a * c) ^(1/2))/2

x2 = (-b - (b^2 – 4 * a * c) ^(1/2))/2

print(“x1 = “, x1)

print(“x2 = “, x2)
```