// ARNAV SINGH
arsingh@jacobs-university.de

Problem 1.1

a.

T= A B B A B B C A C C A B A B A C B C C A B A B

P=A B A b

A b a b

A b a b

A B A b

A B a b

A b a b

A b a b

A B a b

A b a b

A b a b

A B A B

11 alignments, 17 comparisons

b.

T= A B B A B B C A C C A B A B A C B C C A B A B

P= A B A B

a b a B

A B A B

a b A B

a b a B

a b a B

a B A B

A B A B

7 alignments

16 comparisons

c.

T= A B B A B B C A C C A B A B A C B C C A B A B

P= A B A b

a b a B

A B A B

a b A B // good suffix

a b a B

a b a B

a b a B

a b a B

a B A B

A B A B // good suffix 2

9 alignments
16 comparisons

d

T= A B B A B B C A C C A B A B A C B C C A B A B
P= A B A B

```
  a b a B
    A B A B                  // bad character
      a b a B              // good suffix 2
            a b a B  //  bad character
              a B A B    // bad character
                 A B A B  // good suffix 2
```

6 alignments
14 comparisons

Problem 1.2

a.

```
 (-)((*)((+)((*)((+)4 3)2)5)5) ((+)((*) 5 10)3)
```

b.

```
 (42 'div'2) 'gcd' (30 'mod'16)
```

Problem 1.3

a.

PrecedenceLevel	Operator	Associativity
0	$	Right
2	||	Right
3	&&	Right
6	+,-	Left
7	*,/	Left
8	^	Right

b. Since the operater is non associative the programm will give out an error
because it woudn't know what to do with it.

For example,
variable x== variable y>= variable z
In this case haskell doesnt know how to execute this expression above, therfore it will spit out an error.