file =

```
file = open(f->read(f, String), "day_2.txt")
```

```
begin
    #separing each line
    input = eachmatch(r"(\d+)-(\d+) (\w): (\w+)\n",file)

    #getting first match of regex and converting to int
    input = map(x-> x.captures, input)
end
```

```
[x[1] for x in input]
```

```
function valid_line(min ,max ,password, letter::AbstractString)
    min = parse(Int64, min)
    max = parse(Int64, max)

    matchs = collect( eachmatch( Regex(letter) , password) )
    return size(matchs)[1] ≥ min && size(matchs)[1] ≤ max
end
```

resp1 =

```
resp1 = sum([valid_line(x[1],x[2],x[4],x[3]) for x in input])
```

Problem 2

```
md"# Problem 2"
```

```
function valid_line2(pos1 ,pos2 ,password, letter::AbstractString)
    pos1 = parse(Int64, pos1)
    pos2 = parse(Int64, pos2)

    matchs =  collect(eachmatch( Regex(letter) , password))
    if size(matchs)[1] == 0
        return false
    end
    apperance= map(x -> x.offset, matchs)
    return xor((pos1 in apperance) ,(pos2 in apperance))
end
```

resp2 =

```
resp2 = sum([valid_line2(x[1],x[2],x[4],x[3]) for x in input])
```