Regular Expression & Sub Routine

1. BasicRegex.pl

```
#!/usr/bin/perl -w

    use strict;

    my %chars;

    my $Para =

    "Sunflowers waiting for sunshine. \n

    Violets just waiting for dew. \n

    Bees just waiting for honey \n

    And honey, I'm just waiting for you!";
```

```
# Matches 'for'

 print "Matched \n"

     if ($Para =~ m/for/);

# Matches 'And' at start of string

 print "Does not match \n"

     if ($Para =~ m/^And/);

# Matches 'And' at start of each line

 print "Matches (using modifiers) \n"

     if ($Para =~ m/^And/m);
```

1. BasicRegex.pl

```
my $group = "abcd";

   # Grouping captures matched strings

   #           1 2   3  4

   $group =~ m/(a(b|c)(c(d)))/;

   print "$1, $2, $3, $4 \n";

   # Converts lowercase alphabets range [a to m]

   # to uppercase

   $Para =~ tr/[a-m]/[A-M]/;

   print "$Para \n";
```

```
# Counts frequency of uppercase alphabets

 $Para =~ s/([A-Z])/$chars{$1}++;$1/eg;

 print "Frequency of '$_' : $chars{$_} \n"

     foreach (sort{$chars{$b} <=> $chars{$a}}

     keys %chars);

# If we use "m/avi/" then all four words will

# be matched - not GOOD!

# lookahead(?=) and lookbehind(?<=) to match

 my $x = "tavi avi pavi a-avi";

 print "Found avi"

     if ($x =~ m/(?<=\s)avi(?=\s)/);
```

2. Match_contact.pl

```
#!/usr/bin/perl -w-
use strict;
# RegEx using DEFINE block
my $reg = qr/^(?&first_name)[\s]*(?&last_name)[\s]*(?&phone_number)$
   (?(DEFINE)
       (?<first_name>      # first_name matches any number of alphabets
           ([a-zA-Z]+))
       (?<last_name>
           ([a-zA-Z]+))    # first_name matches any number of alphabets
       (?<phone_number>
           ((\+?(\d{1,3})))?    # matches country code if present
           (\-)?                # matches "-" between country code and number
           (\d{10})))           # matches 10 digit for a valid phone number
   )/xn; # used modifier x for free spacing
```

2. Match_contact.pl

```
# Test contacts hash
my %contact = (
        "cont1" => "John Doe +81-9876543210",
        "cont2" => "John +81-9876543210",
        "cont3" => "John Doe +81-123",
        "cont4" => "Jo123hn Doe +81-9876543210",
        "cont5" => "John Doe +819876543210",
        "cont6" => "John   Doe   9876543210",
    );
# validating each contact
foreach my $key (keys %contact){
    print " $key : $contact{$key} is ";
    if ($contact{$key} =~ $reg) {
        print "valid \n";
    }
    else {
        print "invalid \n";
    }
}
```

3. Freq_of_chars.pl

```
#!/usr/bin/perl -w
use strict;
my %chars;
my $Para =
"Sunflowers waiting for sunshine. \n
Violets just waiting for dew. \n
Bees just waiting for honey \n
And honey, I'm just waiting for you!";


print "Matched \n"
   if ($Para =~ m/for/);
print "Does not match \n"
   if ($Para =~ m/^And/);
print "Matches (using modifiers) \n"
   if ($Para =~ m/^And/m);
```

3. Freq_of_chars.pl

```
my $group = "abcd";
$group =~ m/(a(b|c)(c(d)))/;
print "$1, $2, $3, $4 \n";
$Para =~ tr/[a-m]/[A-M]/;
print "$Para \n";

$Para =~ s/([A-Z])/$chars{$1}++;$1/eg;
print "Frequency of '$_' : $chars{$_} \n"
   foreach (sort{$chars{$b} <=> $chars{$a}} keys %chars);


my $x = "tavi avi pavi a-avi";
print "Found avi"
   if ($x =~ /(?<=\s)avi(?=\s)/);
```

4. substitution_regex.pl

```
#!/usr/bin/perl -w

# String in which text

$string = "Hello all!!! Welcome here";

$string =~ s/here/to Polban/;

print "$string\n";
```

5. transliterate_01.pl

```
use strict;
use warnings;
use 5.010;

my $text = 'abc bad acdf';
say $text;

$text =~ tr/a/z/;
say $text;
```

6. Password Validation

Create the regular expression in PERL to validate password text that does the following:

- No white-space character
- Minimum length of 10 characters
- Make sure at least :
  - One uppercase letter
  - One lowercase letter
  - One symbolic letter (non alpha)
  - One numeric letter

7. Domain Web Validation

Create the regular expression to validate URL (domain), consist of (.com, .id, .net), for example :

- www.google.com → valid
- google.com → invalid
- www.google.my → invalid
- yahoo.com → invalid
- www.yahoo.com → valid
- www.abc123.id → valid
- www.abc123.sg → invalid
- www.detik.net → valid

8. Serial Number Matching

Create the regular expression to validate serial number below :

22-Ab627-0360XY → valid

50-Yz6AA-076cUg → valid

- Format : XX-XXXXX-XXXXXX
- Must be exactly 15 characters
- Must only contains number for first 2 digit
- Allowed to use alpha numerics for the rest of digit
- Allowed to use uppercase or lowercase letters

9. Sorting a List

Create the Subroutine to Sorting an arrays , with criteria:

Letter appear first before number

Uppercase appear first before lowercase

Ex:

Input `("Alamat","aku","Alamat04","Handphone","03","02","z9");`

Output: Alamat Alamat04 Handphone aku z9 02 03

10. Add Two Arrays

Create the Subroutine to add elements of two arrays and print the result

```
input
@a = (1, 2, 3);
@b = (4, 5, 6);
Output (5, 7, 9)
```

11.Returned Value From SubRoutine

Create the Subroutine to calculate the average of elements in arrays and print the result

input:

```
$a = calc_avg(11, 20, 2, 8, 3);
$a = calc_average(1,2,3);
```

12.Max - Min

Create two subroutines min() and max() which accept an array as input and calculate the minimum and maximum numeric value of their arguments respectively.

13.Word And Character Count

Create the Subroutine to calculate the total of Word in line and count the character and print the result

Input = JTK Polban adalah sekolah vokasi

Output

Word = 5

Character = 28