

Examination questions for BINP13, 2014-10-31 (09.00 - 13.00).

Approximately 15p are required for passing the exam.

Part 1: Interpret Perl Code

Question 1 (3p): Create the following scalars:

- `$var1`: A reference to the named array `@arr`
- `$var2`: A reference to the anonymous hash consisting of the keys-values pairs: 'one'-3432 and 'Perl'-'Fun'
- `$var3`: A reference to then named hash `%tjohelj`

Question 2 (1p): What is the output of the following program?

```
#!/usr/bin/perl -w
use strict;

my $var1 = 'Perl';
my $var2 = "$var1 is ";
my $var3 = $var2 . "really \"fun\"!";

print "$var3\n";
```

Question 3 (2p): What is the output of the following program?

```
#!/usr/bin/perl -w
use strict;

my $line1 = 'RGT+456 S453DFR 564++AD45QWE4 123456G 5f;;;RT 45()23';
my $line2 = '---C456      ----IA23232AN----- Q-----1454436565555';

my $cnt1 = 0;
while( $line1 =~ /[A-Z]{2,3}\s?\d+/g ) {
    $cnt1++;
}
print "$cnt1\n";

$line2 =~ s/(-|\s|A\d+A|[456])//g;
$line2 =~ tr/CQ/BP/;
print "$line2\n";
```

Question 4 (2p): What is the output of the following program?

```
#!/usr/bin/perl -w
use strict;

my @nums = (3,-3,5,8);

my $val = shift @nums;
my $i = 1;
while ($i <= $val) {
    push @nums, $i;
    $i++;
}

my $cnt = 0;
foreach my $val (@nums) {
    $cnt += $val;
}
print "$cnt\n";
```

Question 5 (2p): What is the output of the following program?

```
#!/usr/bin/perl -w
use strict;

my $str = 'xxxxxI-A-L-GG-I-TWW-KE-A-P-GG-ER-TTA-Lyyyyyy';
$str =~ s/^[xy]+//;
$str =~ s/[xy]+$//;
$str =~ s/-A-/ /g;

my @parts = split /-G\w-|-T\w{2}-/, $str;

my $res = join ' ', @parts;
print "$res\n";
```

Part 2: Write Perl code

Question 6 (3p): You have a nucleotide sequence defined in the scalar `$seq`. Write **Perl code** that replace all codons 'TAG', 'TAA' and 'TGA' with the character '-' and print the number of replacements done.

Question 7 (3p): The Unix command `grep` can be used to search for text in files. As an example,
`>> grep sub parse.pl`

will print all **lines** in the file 'parse.pl' that contains the string 'sub'.

Write a **Perl program** that works like the Unix command `grep`. The first argument to the Perl program should be the string to search for and the second argument should be the file to search in.

Question 8 (3p): Write **Perl code** that calculates the sum

$$\sum_{k=1}^{100} \frac{1}{k^2} = (1 + 1/4 + 1/9 + 1/16 + \dots + 1/10000)$$

and prints it on the screen.

Question 9 (3p): Write a **subroutine** that takes a reference to a hash as the argument. We can assume that all values in the supplied hash are numbers. Return a reference to an array containing all keys in the supplied hash that have values in the interval [0,10].

Question 10 (3p): Write **Perl code** that reads a scalar from standard input. We can assume that this scalar contains a sequence of amino acid letters. Find the number of occurrences of the letter D in this sequence. You should solve this task using the builtin function `index` and not using regular expressions. The documentation for `index` follows here:

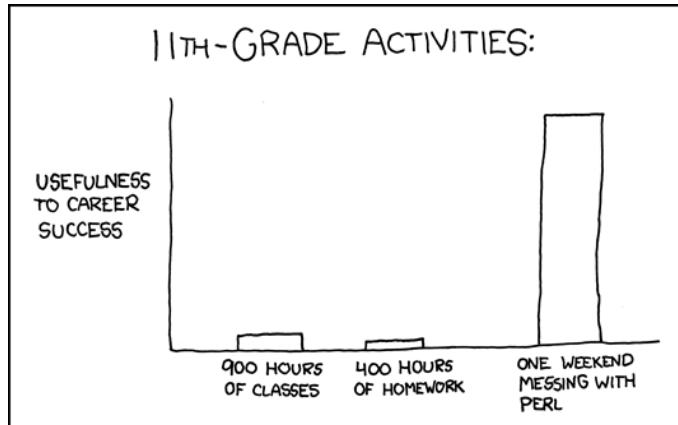
```
index STR,SUBSTR,POSITION  
index STR,SUBSTR
```

The `index` function searches for one string within another, but without the wildcard-like behavior of a full regular-expression pattern match. It returns the position of the first occurrence of `SUBSTR` in `STR` at or after `POSITION`. If `POSITION` is omitted, starts searching from the beginning of the string. `POSITION` before the beginning of the string or after its end is treated as if it were the beginning or the end, respectively. `POSITION` and the return value are based at zero. If the substring is not found, "index" returns -1.

Good Luck!

/Mattias

Usefulness of Perl (<http://xkcd.com/519/>)



The Matrix revealed

