

Pattern matching revision (1)

Suppose `$s = "abcde"`. Do the following patterns match and, if so, what are the values of the match variables (`$1`, `$2`, etc.)?

```
$s =~ /(.*);  
$s =~ /^(.*)/;  
$s =~ /(.*$)/;  
$s =~ /^(.*)$/;  
$s =~ /(..)/;  
$s =~ /(..)(.)/;  
$s =~ /(..)*(.*)/;  
$s =~ /((.)(.))/;  
$s =~ /(((.)(.)))/;  
$s =~ /(.{3})/;  
$s =~ /(.{,3})/;  
$s =~ /(.{3,})/;
```

bigendian.pl

```
#!/usr/bin/perl  
  
open(MAILRC, "$ENV{'HOME'}/.mailrc") ||  
die "Can't open $ENV{'HOME'}/.mailrc: $!\n";  
  
while (<MAILRC>) {  
    chop;  
    if ($_ =~ /^alias.*@/) {  
        ($front, $path) = split(/@/, $_);  
        @parts = split(/\./, $path);  
        if ($parts[0] =~ /^(uk|de|be|com|gov)$/) {  
            print $front, '@', join('.', reverse(@parts)), "\n";  
        } else {  
            print $front, '@', $path, "\n";  
        }  
    } else {  
        print $_, "\n";  
    }  
}
```

Pattern matching revision (2)

Suppose `$s = "abcde"`. What is the value of `$s` after each of the following statements?

```
$s =~ s/./x/;  
$s =~ s/./x/g;  
$s =~ s/.*x/;  
$s =~ s/^.*/x/;  
$s =~ s/.*x/;  
$s =~ s/./x/;  
$s =~ s/./x/g;  
$s =~ s/((.)(.))/321/;  
$s =~ s/((.)(.))/321$2$1/;  
$s =~ s/((.)(.))/321/;  
$s =~ s/[bc]/x/;  
$s =~ s/[bc]/x/g;  
$s =~ s/[bc]+/x/;  
$s =~ s/[bd]+/x/;  
$s =~ s/[bc]*/x/;
```

reverse_complement.pl

```
#!/usr/bin/perl  
  
$dna = "ATGAAT";  
  
$dna =~ tr/ACGT/TGCA/;  
  
$dna = reverse($dna);  
print "Reverse complement: ", $dna, "\n";
```

palindrome.pl

```
#!/usr/bin/perl  
  
while ( <> ) {  
    chomp;  
    if ( $_ eq reverse($_) ) {  
        print "$_ is a palindrome\n";  
    }  
}
```

embl_orf.pl

```
#!/usr/bin/perl

$sequence = "";

while ( <> ) {
    if ( /^ / ) {
        s/[^a-z]//g;
        $sequence .= $_;
    }
}

while ( $sequence =~ /(atg.*)/ ) {
    $orf = $1;
    $sequence = substr($1, 3);
    $orf =~ s/(...)\1/g;
    $orf =~ s/(taa|tag|tga).*/;
    $orf =~ s//g;
    print "$orf\n";
}
```

Graham Kemp, Chalmers University of Technology

equal.pl

```
#!/usr/bin/perl

print "Type in a string (e.g. aaapaaeaaaaa): ";
$s=<STDIN>;
chomp($s);
print "$s\n";

if ( $s =~ /^(a*)p(a*)e(a*)$/ &&
    length($1) + length($2) == length($3) ) {

    print "match\n";
} else {
    print "don't match\n";
}
```

An alternative test

```
if ( $s =~ /^(a*)p(a*)e\1\2$/ ) {
```

Graham Kemp, Chalmers University of Technology

translate.pl

```
%standardgeneticcode = (
    "ttt"=> "Phe", "tct"=> "Ser", "tat"=> "Tyr", "tgt"=> "Cys",
    ....
    "gtg"=> "Val", "gcg"=> "Ala", "gag"=> "Glu", "ggg"=> "Gly");

while ($line = <DATA>) {
    print "$line";
    chomp($line);
    $line =~ s/(...)\1/g;
    @triplets = split(" ", $line);
    foreach $codon (@triplets) {
        print "%standardgeneticcode{$codon}";
    }
    print "\n\n";
}

__END__
atgcatccctttaat
tctgtctga
```

Graham Kemp, Chalmers University of Technology

assemble.pl

Fragments:

```
rs International Mas
onal Mas
ernational Masters Prog
me in Bio
Bioinformatics
Chalmers Interna
rs Programme in Bio
```

Assembled sequence:

```
Chalmers International Masters Programme in Bioinformatics
```

Graham Kemp, Chalmers University of Technology

my_dotplot.pl (part 1)

```
#!/usr/bin/perl

#
# file:      my_dotplot.pl
# purpose:   read two SWISS-PROT files and produce a dotplot
#

open(SWISSPROT_FILE, $ARGV[0]) || die "Can't open $ARGV[0]: $!\n";
$seq1 = "";
while ( <SWISSPROT_FILE> ) {
    chomp;
    if ( /^      / ) {
        s/ //g;
        $seq1 .= $_;
    }
}
close SWISSPROT_FILE;
```

Graham Kemp, Chalmers University of Technology

my_dotplot.pl (part 2)

```
open(SWISSPROT_FILE, $ARGV[1]) || die "Can't open $ARGV[1]: $!\n";
$seq2 = "";
while ( <SWISSPROT_FILE> ) {
    chomp;
    if ( /^      / ) {
        s/ //g;
        $seq2 .= $_;
    }
}
close SWISSPROT_FILE;

for ( $i=0 ; $i<length($seq1) ; $i++ ) {
    for ( $j=0 ; $j<length($seq2) ; $j++ ) {
        if ( substr($seq1,$i,1) eq substr($seq2,$j,1) ) {
            print "$i $j\n";
        }
    }
}
```

Graham Kemp, Chalmers University of Technology