

Module Programmation Internet.

Correction TD 3

Damien Sauveron

Exercice 1.

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

my $somme = 0;

foreach $valeur (@ARGV)
{
    $somme += $valeur;
}

print "$somme\n";
```

Exercice 2.

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

@ARGV || die "Fournissez au moins un argument au script";

@lettres = grep { lc ge 'a' && lc le 'z' } (split //,$ARGV[0]);

@chiffres = grep { $_ ge '0' && $_ le '9' } (split //,$ARGV[0]);

print "Nombre de lettres : ",scalar(@lettres),"\n";

print "Nombre de chiffres : ",scalar(@chiffres),"\n";
```

Exercice 3.

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

$chaine = <STDIN>;
chomp $chaine;
if ($chaine eq reverse($chaine))
{
    print "$chaine est un palindrome.\n";
}
else
{
    print "$chaine n'est pas un palindrome.\n";
}
```

Exercise 4.

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

$n = <STDIN>;
chomp $n;

print "\n";

for ($indice = 0; $indice <= $n; $indice++)
{
    print "2**$indice = ", 2**$indice, "\n";
}
```

Exercise 5.

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

sub max
{
    @_ || return undef;
    my $maximum = shift;
    foreach $x (@_)
    {
        $maximum = $x if ($x > $maximum);
    }
    return $maximum;
}

print "<< undef >>\n" if (!defined(max()));

print max(1,8,2,5), "\n";

my @t = (2,0,-6,'aa',7,-78);
print max(@t), "\n";

print max(2,0,6,97.2,0,-78.5), "\n";
```

Exercise 6.

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

sub puissances
{
    @_ || return 2**0;

    my $n = shift;
    my @resultat = ();

    foreach $indice (0..$n)
    {
        push (@resultat, 2**$indice);
    }
    return @resultat;
}

@r = puissances(3);
print "@r\n";

@r = puissances();
```

```

print "@r\n";

@r = puissances 5;
$" = ','; # Specifie le caractere utilise entre chaque element d'une liste
          # lors d'une interpolation de celle-ci dans une chaine.
print "@r\n";

```

Exercice 7.

```

#!/usr/bin/perl -w

sub remplacer
{
    my $pos = index $_[0], $_[1];
    if ($pos != -1)
    {
        substr($_[0], $pos, length($_[1])) = $_[2];
    }
}

my $s = "Le chat mange";

remplacer($s, "chat", "chien");
print "$s\n";

remplacer($s, "", "Oui : ");
print "$s\n";

remplacer($s, "loup", "chat");
print "$s\n";

```

Exercice 8.

```

#!/usr/bin/perl -w

sub creer_identite
{
    my %matrice = ();
    my $n = shift;

    foreach $ligne (1..$n)
    {
        foreach $colonne (1..$n)
        {
            if ($ligne == $colonne)
            {
                $matrice{"$ligne:$colonne"} = 1;
            }
            else
            {
                $matrice{"$ligne:$colonne"} = 0;
            }
        }
    }

    return %matrice;
}

```

```

sub classement
{
    ($i1,$j1) = split /:/,$a;
    ($i2,$j2) = split /:/,$b;
    return -1 if ($i1 < $i2);
    return -1 if ($i1 == $i2 && $j1<$j2);
    return 0  if ($i1 == $i2 && $j1==$j2);
    return 1  if ($i1 == $i2 && $j1>$j2);
    return 1;
}

sub afficher
{
    my %matrice = @_;

    @cles = keys %matrice;
    @cles = sort classement @cles;

    foreach $cle (@cles)
    {
        ($i,$j) = split /:/,$cle;
        print "Matrice($i,$j) = $matrice{$cle}\n";
    }
}

my %I = creer_identite(3);
afficher(%I);

print "\n";

%I = creer_identite(2);
afficher(%I);

```