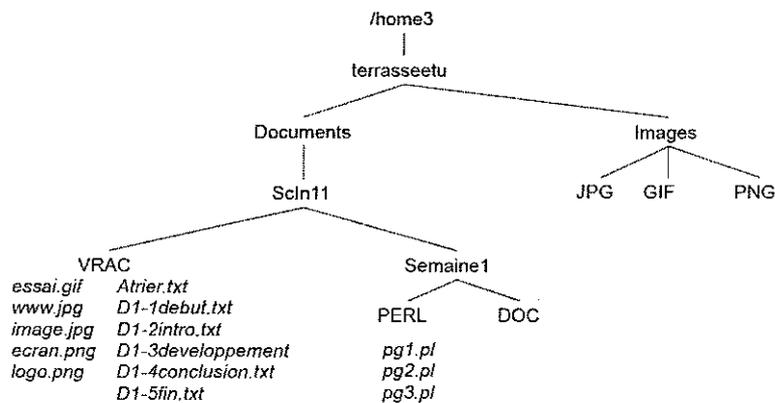


Vous pouvez répondre sur l'énoncé

Les différents fichiers de données et programmes sont donnés en annexe

Exercice 1 :



Le résultat de la commande `ls -l` dans le répertoire `PERL` est le suivant :

```
-r--r--r-- 1 terrasseetu terrasseetu 511 27 juin 12:52 pg1.pl
-r-xr-xr-x 1 terrasseetu terrasseetu 477 27 juin 12:46 pg2.pl
-r-xr-xr-x 1 terrasseetu terrasseetu 1240 28 juin 08:20 pg3.pl
```

- 1) Vous travaillez dans le répertoire `DOC`, quelle instruction doit être utilisée pour vous déplacer dans le répertoire `VRAC` ?
- 2) Vous travaillez dans le répertoire `PERL`, indiquez la (les) instruction(s) nécessaire(s) pour exécuter le programme `pg1.pl`
- 3) Vous travaillez dans le répertoire `VRAC`. Quelle instruction doit être utilisée pour copier dans le répertoire `JPG` tous les fichiers d'extension `jpg` ?

- 4) Vous travaillez dans un répertoire qui contient une archive `ecrans.zip`. Vous devez l'envoyer dans le répertoire `Images` du compte de `terrasseetu`. Vous avez fait un `sftp terrasseetu@ufrsciencestech.u-bourgogne.fr` et vous êtes placé dans le répertoire VRAC du compte `terrasseetu`. Indiquez la (les) instruction(s) nécessaire(s) pour envoyer l'archive.

Exercice 2 :

Vous disposez d'un fichier `couleurs` donné en annexe 1 décrivant des couleurs par un nom et quatre codages (RVB en hexadécimal et en décimal, CMJN et TSL) :

- les lignes commençant par `#` sont des commentaires,
- le nom de la couleur est placé en début de ligne et entre guillemets,
- le codage RVB (rouge vert bleu) est constitué de trois valeurs comprises :
 - entre 00 et FF, en hexadécimal,
 - entre 0 et 255, en décimal,
- le codage CMJN (cyan, magenta, jaune, noir) est constitué de quatre valeurs entre 0 et 100,
- le codage TSL (teinte, saturation, luminosité) est constitué de trois valeurs comprises:
 - entre 0 et 360 pour la première,
 - entre 0 et 100 pour les deux autres,
- toutes les valeurs sont séparées par un espace.

1) On exécute le programme `progl1.pl` donné en annexe. Quelles lignes seront affichées ?

2) On considère le programme `progl2.pl`. Quel sera le résultat de l'exécution de ce programme ? (donnez le résultat uniquement pour les 2 premières couleurs Abricot et Absinthe)

3) Questions sur le programme prog2c.pl

a. Quel est le résultat de l'exécution du programme prog2c.pl ? (donnez le résultat uniquement pour les 2 premières couleurs)

b. Quelle instruction faut-il rajouter

i. pour remplacer toutes les parenthèses fermantes par un saut de ligne ?

ii. Pour remplacer toutes les parenthèses ouvrantes par un espace ?

Annexe 1 : Données et programmes de l'exercice 2

Données exercice 2 : fichier couleurs

```
# RVB hexa - RVB decimal - CMJN - TSL
# RVB (ordre des valeurs : rouge vert bleu)
# CMJN (ordre des valeurs : cyan, magenta, jaune, noir)
# TSL (ordre des valeurs : teinte, saturation, luminosité)
"Abricot" E6 7E 30 230 126 48 5 60 87 0 26 79 90
"Absinthe" 7F DD 4C 127 221 76 43 0 66 13 99 66 87
"Acajou" 88 42 1D 136 66 29 0 51 79 47 21 79 53
"Aigue-marine" 79 F8 F8 121 248 248 51 0 0 3 180 51 97
"Aile de corbeau (cheveux)" 00 00 00 0 0 0 0 0 0 0 100 0 0 0
"Albâtre" FE FE FE 254 254 254 0 0 0 0 0 0 100
"Alezan (chevaux)" A7 67 26 167 103 38 0 38 77 35 30 77 65
"Amande" 82 C4 6C 130 196 108 34 0 45 23 105 45 77
"Amarante" 91 28 3B 145 40 59 0 72 59 43 349 72 57
"Ambre jaune" F0 C3 00 240 195 0 0 19 100 6 49 100 94
"Ambre rouge" AD 39 0E 173 57 14 0 67 92 32 16 92 68
"Améthyste" 88 4D A7136 77 167 19 54 0 35 279 54 65
"Anthracite" 30 30 30 48 48 48 0 0 0 81 0 0 19
"Aquilain (chevaux)" AD 4F 09 173 79 9 0 54 95 32 26 95 68
```

Programme prog1a.pl

```
#!/usr/bin/perl
use strict;
use warnings;

{
    my $lec;
    my $fich1="couleurs";
    my $ligne;

    open ( $lec, "<", $fich1 ) or die "erreur sur $fich1";

    while ( <$lec> )
        { $ligne=$_;

            if ( $ligne =~ /[A-Za-z]+/ )
                {
                    print "$ligne";
                }
        }

    close($lec);
} # fin du programme
```

Programme prog1b.pl

```
#!/usr/bin/perl
use strict;
use warnings;

{
  my $lec;
  my $ecr;
  my $fich1="couleurs";
  my $fich2="coulRGB";
  my $ligne;

  open ( $lec, "<", $fich1 ) or die "erreur sur $fich1";
  open ( $ecr, ">", $fich2 ) or die "erreur sur $fich2";

  while ( <$lec> )
    { $ligne=$_;

      if ( $ligne =~ /^"(.*)" ([A-Z0-9]{2} [A-Z0-9]{2} [A-Z0-9]{2}) ([0-9]{1,3} [0-9]{1,3} [0-9]{1,3})/ )
        {
          print $ecr "$1 \n hexa : $2 \n dec. : $3 \n";
        }
    }

  close($lec);
} # fin du programme
```

Annexe 2 : Données et programmes de l'exercice 3

Données exercice 3 : fichier coul

Honeydew (web color) (Hex: #F0FFF0) (RGB: 240, 255, 240)
Erin (Hex: #00FF3F) (RGB: 0, 255, 63)
Spring Green (web color) (Hex: #00FF7F) (RGB: 0, 255, 127)
BLUE (Hex: #0000FF) (RGB: 0, 0, 255)
GREEN (web color "Lime") (Hex: #00FF00) (RGB: 0, 255, 0)
RED (Hex: #FF0000) (RGB: 255, 0, 0)
Aqua (Hex: #00FFFF) (RGB: 0, 255, 255)
Turquoise Blue (Hex: #00FFEF) (RGB: 0, 255, 239)

Programme prog2a.pl

```
#!/usr/bin/perl
use strict;
use warnings;

{

    my $lec;
    my $ecr;
    my $fich1="coul";
    my $fich2="res";
    my $ligne;

    open ( $ecr, ">", $fich2 ) or die "erreur sur $fich2";
    open ( $lec, "<", $fich1 ) or die "erreur sur $fich1";

    while ( <$lec> )
    { $ligne=$_;
      if ( $ligne =~ / ZONE 1 / )
      {
          print $ecr "$ligne";
      }
    }

    close($lec);
    close($ecr);
} # fin du programme
```

Programme prog2b.pl

```
#!/usr/bin/perl
use strict;
use warnings;

{
    my $lec;
    my $fich1="coul";
    my $ligne;

    open ( $lec, "<", $fich1 ) or die "erreur sur $fich1";

    while ( <$lec> )
        { $ligne=$_;
          if ( $ligne =~ /\((.*)\)/ )
            {
              print "$1 \n";
            }
          }
    close($lec);
} # fin du programme
```

Programme prog2c.pl

```
#!/usr/bin/perl
use strict;
use warnings;

{
    my $lec;
    my $secr;
    my $fich1="coul";
    my $fich2="res";
    my $ligne;

    open ( $secr, ">", $fich2 ) or die "erreur sur $fich2";
    open ( $lec, "<", $fich1 ) or die "erreur sur $fich1";

    while ( <$lec> )
        { $ligne=$_;

          if ( $ligne =~ /^[A-Za-z]+ / )
            {
              $ligne=~s/\(\/:\n/;

              print $secr "$ligne";
            }
          }
    close($lec);
    close($secr);
} # fin du programme
```