```ruby
require_relative "regles.rb"
require_relative "eleve.rb"

#----Méthodes auxiliaires
def lire_fichier(nom_fichier)
    # Créer plusieurs objets Eleve lus du fichier s1.csv,
    # Mettre en mémoire les objets Eleve dans le Array eleves,
    eleves = Array.new #Création d'un objet Array vide
    f = File.open(nom_fichier, "r")
    while ligne = f.gets
        ligne = ligne.chomp
        infos = ligne.split(";")
        params = {  :numero => infos[0],
                    :nom => infos[1],
                    :prenom => infos[2],
                    :sexe => infos[3],
                    :programme => infos[4],
                    :natation => infos[5],
                    :anglais => infos[6],
                    :musique => infos[7]
        }
        eleves << Eleve.new(params)
    end
    f.close
    return eleves
end

def afficher(liste)
    liste.each {|element| puts element} if liste.class == Array
    liste.each {|cle, valeur| puts "#{cle}\t#{valeur}"} if liste.class == Hash
end

def compter_eleves_dans_le_groupe(groupe, eleves)
    return eleves.find_all{|eleve| eleve.groupe == groupe}.size
end

def compter_eleves_dans_le_groupe_anglais(cours, eleves)
    return eleves.find_all{|eleve| eleve.cours_ang == cours}.size
end

def compter_eleves_dans_le_groupe_musique(cours, eleves)
    return eleves.find_all{|eleve| eleve.cours_musique == cours}.size
end

def compter_eleves_de_tous_groupes(groupes, eleves)
    total = Hash.new
    groupes.each do |groupe|
        total[groupe] = compter_eleves_dans_le_groupe(groupe, eleves)
    end
    return total
end

def compter_eleves_de_tous_groupes_anglais(cours_ang, eleves)
```

```ruby
    total = Hash.new
    cours_ang.each do |cours|
        total[cours] = compter_eleves_dans_le_groupe_anglais(cours, eleves)
    end
    return total
end

def compter_eleves_de_tous_groupes_musique(cours_musique, eleves)
    total = Hash.new
    cours_musique.each do |cours|
        total[cours] = compter_eleves_dans_le_groupe_musique(cours, eleves)
    end
    return total
end

def valider?(totaux, maximas)
    return totaux.all?{|groupe, total| total <= maximas[groupe]  }
end

def choisir_eleve_au_hasard(eleves)
    return eleves[rand(eleves.size)]
end

def ecrire_fichier(nom_fichier, eleves)
    f = File.open(nom_fichier, "w")
    eleves.sort{|a,b| a.groupe <=> b.groupe}.each{|eleve| f.write(eleve.to_s + "\n")}
    f.close
end

#-----Programme principal
Eleve.regles(S2)
Eleve.ang(Ang2)
Eleve.mus(Mus2)

max_reg = 31
max_enr = 34

max_eleves_par_groupe = {
    "21" => max_reg,
    "23" => max_reg,
    "25" => max_reg,
    "27" => max_reg,
    "28" => max_reg,
    "29" => max_reg,
    "22" => max_enr,
    "24" => max_enr,
    "26" => max_enr,
}

groupes = max_eleves_par_groupe.keys

max_eleves_par_groupe_ang = {
    "ELA204-00001" => 36,
```

```ruby

    "EESL204-00001" => 35,
    "EESL204-00003" => 35,
    "EESL204-00005" => 35,
    "EESL204-00007" => 35,

    "ESL204-00003" => 31,
    "ESL204-00005" => 31,
    "ESL204-00007" => 31,
    "ESL204-00029" => 31,
}

cours_ang = max_eleves_par_groupe_ang.keys

max_eleves_par_groupe_mus = {
    "MUG204-00001" => 34,
    "MUA204-00001" => 34,

    "MUG204-00003" => 34,
    "MUA204-00003" => 35,

    "MUG204-00025" => 33,
    "MUG204-00026" => 33,
    "MUG204-00027" => 33,
    "MUG204-00028" => 33,
    "MUG204-00029" => 33,
}

cours_mus = max_eleves_par_groupe_mus.keys

eleves = lire_fichier("s2.csv")

puts "Répartition initiale"
eleves.each {|eleve| eleve.assigner_groupe_au_hasard }

totaux_des_groupes = compter_eleves_de_tous_groupes(groupes, eleves)
totaux_des_groupes_ang = compter_eleves_de_tous_groupes_anglais(cours_ang, eleves)
totaux_des_groupes_mus = compter_eleves_de_tous_groupes_musique(cours_mus, eleves)

puts "GROUPES"; afficher(totaux_des_groupes)
puts "ANGLAIS"; afficher(totaux_des_groupes_ang)
puts "MUSIQUE"; afficher(totaux_des_groupes_mus)

fini = false
compteur = 0; max_compteur = 100000

while !fini and compteur < max_compteur
    compteur = compteur + 1

    eleve = choisir_eleve_au_hasard(eleves)
    groupe = eleve.groupe
    groupe_anglais = eleve.cours_ang
    groupe_musique = eleve.cours_musique
```

```
160
161       nombre_eleves = compter_eleves_dans_le_groupe(groupe, eleves)
162       nombre_eleves_ang = compter_eleves_dans_le_groupe_anglais(groupe_anglais, eleves)
163       nombre_eleves_mus = compter_eleves_dans_le_groupe_musique(groupe_musique, eleves)
164
165       if  nombre_eleves > max_eleves_par_groupe[groupe] or
166           nombre_eleves_ang > max_eleves_par_groupe_ang[groupe_anglais] or
167           nombre_eleves_mus > max_eleves_par_groupe_mus[groupe_musique]
168       then
169           eleve.assigner_groupe_au_hasard
170
171           totaux_des_groupes = compter_eleves_de_tous_groupes(groupes, eleves)
172           totaux_des_groupes_ang = compter_eleves_de_tous_groupes_anglais(cours_ang,
      eleves)
173           totaux_des_groupes_mus = compter_eleves_de_tous_groupes_musique(cours_mus,
      eleves)
174
175
176           gr = valider?(totaux_des_groupes, max_eleves_par_groupe)
177           ang = valider?(totaux_des_groupes_ang, max_eleves_par_groupe_ang)
178           mus = valider?(totaux_des_groupes_mus, max_eleves_par_groupe_mus)
179           fini =  (gr and ang and mus)
180
181           #~ puts "#{v1}, #{v2} = #{fini}"
182       end
183
184   end
185
186   puts "\nRépartition finale après #{compteur} itérations"
187
188   puts "GROUPES"; afficher(totaux_des_groupes)
189   puts "ANGLAIS"; afficher(totaux_des_groupes_ang)
190   puts "MUSIQUE"; afficher(totaux_des_groupes_mus)
191
192   ecrire_fichier("groupes_s2.txt", eleves)
```