```ruby
def generer_baremes(nom_fichier)
    #Ne pas trop porter attention a cette méthode!
    #Il faut simplement retenir qu'elle produit un Hash
    #qui est composé de Arrays :
    # {"S1M"=>["07:45", "08:30", "09:00", "09:30", "11:00", "12:00", "13:00", "13:30", "14:00", "14:30"],
    #  "S1F"=>["08:15", "09:00", "10:00", "11:00", "12:00", "13:00", "14:00", "15:00", "15:30", "16:00"],
    #   S2M"=>["07:15", "08:00", "08:30", "09:30", "10:30", "11:30", "12:30", "13:00", "13:30", "14:00"],
    #  "S2F"=>["07:45", "08:45", "09:45", "10:45", "11:45", "12:45", "13:45", "14:15", "14:45", "15:15"],
    #  "S3M"=>["11:00", "12:00", "13:00", "14:00", "15:00", "16:00", "16:45", "17:30", "18:15", "19:00"],
    #  "S3F"=>["13:00", "14:00", "15:30", "16:30", "18:00", "19:00", "20:00", "20:45", "21:30", "22:15"],
    #  "S4M"=>["11:00", "12:00", "13:00", "14:00", "15:00", "16:00", "16:45", "17:30", "18:15", "19:00"],
    #  "S4F"=>["13:01", "14:01", "15:31", "16:31", "18:01", "19:01", "20:21", "20:46", "21:31", "22:16"],
    #  "S5M"=>["10:45", "11:45", "12:45", "13:45", "14:45", "15:45", "16:30", "17:15", "18:00", "18:45"],
    #  "S5F"=>["12:45", "13:45", "15:15", "16:15", "17:45", "18:45", "19:45", "20:30", "21:15", "22:00"],
    #  "P5M"=>["06:45", "06:55", "07:40", "08:15", "08:55", "09:25", "09:55", "10:30", "11:05", "11:20"],
    #  "P5F"=>["06:50", "07:00", "07:45", "08:26", "09:05", "09:35", "10:10", "10:45", "11:20", "11:35"],
    #  "P6M"=>["06:20", "06:30", "07:15", "07:50", "08:30", "09:00", "09:30", "10:05", "10:40", "11:00"],
    #  "P6F"=>["06:35", "06:50", "07:30", "08:10", "08:50", "09:20", "10:00", "10:35", "11:10", "11:35"]}

    baremes = {}
    a = []

    f = File.open(nom_fichier,"r")
    while ligne = f.gets
        ligne = ligne.chomp
        data = ligne.split(";")
        a << data
    end
    f.close

    #Ici, je fais des pirouettes pour assimiler
    #les données du fichier du département EPS.
        14.times do |index|
        bareme_niveau_sexe = a[0][index+1]
        baremes[ bareme_niveau_sexe  ]  = []
        10.times do |i|
        baremes[ bareme_niveau_sexe  ]  << a[i+1][index+1]
        end
    end
```

```ruby

    #Retournons le Hash baremes.
    return baremes
end

def obtenir_bareme(niveau, sexe, baremes)
    #On peut additionner des String pour former une autre String.
    bareme_niveau_sexe = (niveau+sexe).upcase

    #Ici nous déclarons une ERREUR si le niveau et le sexe
    #ne sont pas une clé du Hash baremes.
    if !baremes.has_key?(bareme_niveau_sexe)
    then
        # raise affiche le message et arrête l'exécution
        # du programme!
        raise "Erreur! Niveau ou sexe invalide"
    end

    #Récupérons le barème approprié pour cet élève
    #Le barème est un Array
    bareme = baremes[bareme_niveau_sexe]

    return bareme
end

def note_xc(temps, bareme)

    #9:22 devient 09:22 avec la méthode rjust(5,"0").
    #Expérimentez en changeant le 5 et le "0" pour autre chose!
    temps = temps.rjust(5, "0")

    #Trouvons l'index de l'élément du Array bareme
    #qui est plus grand que le temps de cet élève.
    index = bareme.find_index{|x| x > temps }

    #Si aucun index n'est trouvé, alors note = 50.
    if index.nil?
    then
        note = 50
```

```ruby
    else
        #Le barème baisse de 5 points à chaque élément.
        note = 100-index*5
    end

    #Retournons la note demandée.
    return note
end

#====MAIN
#Générons la Hash qui contient les 14 barèmes.
baremes = generer_baremes("baremes.csv")

nom_fichier = "XC-S3-2017.txt"
#Ouvrons le fichier en mode LECTURE : read.
f = File.open(nom_fichier, "r")

while ligne = f.gets
    #Voici comment les 14 éléments d'information sont présentés
    # dans le fichier :

    #Place  Bib Name First name Last name   Team name
    # 0     1   2    3          4           5
    #Category   Info 1  Info 2  Time    Difference  % Back
    # 6         7       8       9       10          11
    #% Winning  % Average
    # 12        13

    #Chomp supprime les caractères de changement de ligne
    #qui nous dérange!
    ligne = ligne.chomp

    # La ligne est séparée en 14 éléments de données qui
    # seront placées dans le Array data.
    # Les éléments sont séparés par une TABULATION "\t".
    data = ligne.split("\t")

    #Donnons un nom siginficatif aux éléments d'information.
    rang    = data[0]
```

```ruby
    dossard = data[1]
    #L'élément data[2] ne nous intéresse pas, alors on l'ignore!
    prenom  = data[3]
    nom     = data[4]
    groupe  = data[5]
    sexe    = data[6]
    sigle   = data[7]

    # 2 devient 00002
    groupe_coba = data[8].rjust(5,"0")

    # 9:42.2 se fait séparer en deux parties qui
    # sont placées dans un Array :  [9:42 , 2].
    temps = data[9].split(".")[0]

    #Le premier caractère du groupe est le niveau.
    niveau  = "S" + groupe[0]

    if temps == "DNF" #DNF = Did not finish race
    then
        note = "DNF"
    else
        #Trouvons la note associée au temps de cet élève
        #selon
        bareme_pour_cet_eleve = obtenir_bareme(niveau, sexe, baremes)
        note = note_xc(temps, bareme_pour_cet_eleve)
    end

    #On affiche les données séparées par un point-virgule car cela se
    #lit facilement avec Excel : fichier CSV .
    puts "#{dossard};#{nom};#{prenom};#{groupe};#{sexe};#{sigle};#{groupe_coba};#{temps};#{niveau};#{note}"
end

#Fermons le fichier.
f.close
```