▶ **HX4 & HX1 – Option Informatique**
Année 2011, Neuvième TP Caml
Louis Jachiet (`http://www.eleves.ens.fr/home/jachiet/`)

Lorem ipso

Solum reperesimunum ali pos pras bellis cum et minemisciennicessenes stritimena.

▶ **Question 1 *** *Récupérer sur internet la fonction texte et réussir à la faire marcher. Récupérer les textes 0 à 7.*

1 Quel langue parlez-vous ?

1.1 Vecteur de fréquence

▶ **Question 2** *Écrire une fonction* ***intchar*** *qui envoie 'a'..'z' sur 0..25*

▶ **Question 3** *Écrire une fonction qui prend un mot et une position et rajoute dans un tableau* ***freq*** *le nombre d'occurrences de chaque lettre. On suppose que le tableau est déclaré à l'extérieur de la fonction.*

▶ **Question 4** *En déduire une fonction* ***frequence*** *qui prend une liste de mot et qui renvoie un tableau comptant le nombre d'occurrences de chaque lettre.*

1.2 Retrouver la langue d'un texte

Pour retrouver la langue des textes, on va se contenter de comparer les fréquences d'apparition de chacune des lettres. Pour cela, il faut être capable de normaliser le tableau renvoyé par **frequence**.

▶ **Question 5** *Écrire une fonction* ***somme2*** *qui prend un tableau d'entiers et renvoie sa norme $\mathcal{L}^2$.*

▶ **Question 6** *Écrire une fonction* ***normalise*** *qui normalise un vecteur donné en entrée.*

▶ **Question 7** *Écrire la fonction* ***distance*** *qui donne le carré de la distance (norme 2) entre deux vecteurs.*

▶ **Question 8** *Écrire une fonction qui prend un ensemble de textes (sous la forme que vous voulez) et affiche les textes qui sont proches (mettons à distance inférieure à 0.1).*

2 Générer des mots

2.1 Trigrammes

▶ **Question 9** *Écrire une fonction qui prend un texte et qui renvoie un tableau tridimensionnel contenant toutes les occurrences de trigrammes apparaissant dans le texte, avec répétition. Les deux premières dimensions servent à caractériser les deux premières lettres des trigrammes tandis que la troisième sert juste à indicer. La troisième lettre du trigramme est la valeur dans le tableau. Par exemple si abc apparait dans notre texte, on aura c dans la case t.(intcar a).(intcar b).(i) pour un certain i. D'autres part, on allonge (virtuellement) les mots par des espaces. Aussi dans le mot eau on a les trigrammes __e, _ea,eau,au_,u__. On pourra commencer par faire une fonction qui prend un mot et rajoute ses trigrammes à un tableau.*

▶ **Question 10** *Écrire une fonction qui affiche un mot au hasard en respectant les probabilités d'apparition des trigrammes d'un texte donné.*

▶ **Question 11** *En déduire une fonction qui affiche un nombre arbitraire de mot.*

▶ **Question 12** *Générer des phrases aléatoirement dans diverses langues.*

3 Pour ceux qui s'ennuient

▶ **Question 13** *Reconnaitre les langues en fonction des trigrammes.*

▶ **HX4 & HX1 – Option Informatique**
Année 2011, Neuvième TP Caml
Louis Jachiet (`http://www.eleves.ens.fr/home/jachiet/`)

Lorem ipso

Un corrigé

▶ Question 2

```
let intchar c =
  if c >= 'a' && c <= 'z'
  then int_of_char c - int_of_char 'a'
  else failwith "n'existe pas"
;;
```

▶ Question 3

```
let rec do_freq s i =
    if i < string_length s
    then t.(intchar s.[i])<-t.(intchar s.[i])+1
  in
```

▶ Question 4

```
let freq y =
  let t = make_vect 26 0 in
  let rec do_freq s i =
    if i < string_length s
    then t.(intchar s.[i])<-t.(intchar s.[i])+1
  in
  let rec foo = function
    | [] -> ()
    | a::q -> do_freq a 0; foo q
  in
foo y ; t
;;
```

▶ Question 5

```
let sum t =
  let rec foo i =
    if vect_length t = i then 0 else t.(i)+foo (i+1)
  in
foo 0;;
```

▶ Question 6

```
let normalize t =
  let tot = (sum2 t) in
  let res = make_vect (vect_length t) 0. in
  let rec foo i =
    if i < vect_length t
    then (res.(i) <- float_of_int t.(i) /. tot ; foo (i+1))
  in
  foo 0 ; res
;;
```

▶ Question 8

```
let proches txt =
  let e = map_vect (fun x -> normalize (freq x)) txt ;;
  for i = 0 to vect_length e - 2 do
    for j = i+1 to vect_length e - 1 do
      if (dist e.(i) e.(j))<0.13 then (print_int i ; print_string " <-> "; print_int
    done;
  done;;
```

▶ Question 9

```
let make_trigramme t =
  let res = make_matrix 27 27 [] in
  let rec do_mot p1 p2 m i =
    let c = if i = string_length m
      then ' '
      else (do_mot p2 (intchar m.[i]) m (i+1);m.[i]) in
    res.(p1).(p2) <-c::res.(p1).(p2)
  in
  let rec do_text = function
    | [] -> ()
    | a::q -> do_mot 26 26 a 0; do_text q
  in
  do_text t ; map_vect (map_vect vect_of_list) res
;;
```

▶ Question 10

```
let rec write_mot t a b =
  let r = random__int (vect_length t.(a).(b)) in
  let c = t.(a).(b).(r) in
  print_char c ;
  if c <> ' ' then write_mot t b (intchar c)
  ;;
```

▶ Question 11

```
let rec write t n =
  if n > 0
  then
    (
      write_mot t 26 26;
      write t (n-1)
    )
  else print_newline()
;;
```