

Test TP OPM
Durée 45 minutes
énoncé et corrigé

Enoncé :

- 1) Ecrire une fonction **suppToutes** supprimant toutes les occurrences d'un caractère dans une chaîne de caractères : **suppToutes** 'p' 'apple' = 'ale'.
- 2) Ecrire une fonction **nocc** comptant le nombre d'occurrences d'un caractère dans une chaîne de caractères : **nocc** 'a' 'usthb'=0, **nocc** 'l' 'Algérie nouvelle'=3.
- 3) Ecrire une fonction **occurrences** transformant une chaîne de caractères en une liste contenant pour chaque caractère c de la chaîne, la paire (c,n), n étant le nombre d'occurrences de c dans la chaîne : **occurrences** 'apple'=[('a',1);('p',2);('l',1);('e',1)].

Indications :

- 1) si x est une chaîne de caractères et i un entier, **x.[i]** donne le caractère de x d'indice i, le premier caractère étant d'indice 0.
- 2) Utiliser les synonymes **lg** et **sc** des fonctions prédéfinies **string_length** et **sub_string** :
let **lg** x = **string_length** x ;;
let **sc** x i j = **sub_string** x i j ;;

Bon courage

Corrigé :

1)

```
let lg x = string_length x;;
```

```
lg : string -> int = <fun>
```

```
let sc x i j = sub_string x i j;;
```

```
sc : string -> int -> int -> string = <fun>
```

```
let rec suppToutes c x =
```

```
  if x="" then ""
```

```
  else let suffixe = sc x 1 (lg x-1) in
```

```
    if c = x.[0] then suppToutes c suffixe
```

```
    else sc x 0 1 ^ suppToutes c suffixe;;
```

```
suppToutes : char -> string -> string = <fun>
```

2)

```
let rec nocc c x =
```

```
  if x="" then 0
```

```
  else let suffixe = sc x 1 (lg x-1) in
```

```
    if c = x.[0] then 1+nocc c suffixe
```

```
    else nocc c suffixe;;
```

```
nocc : char -> string -> int = <fun>
```

3)

```
let rec occurrences x =
```

```
  if x="" then []
```

```
  else (x.[0],nocc x.[0] x)::occurrences (suppToutes x.[0] x);;
```

```
occurrences : string -> (char * int) list = <fun>
```