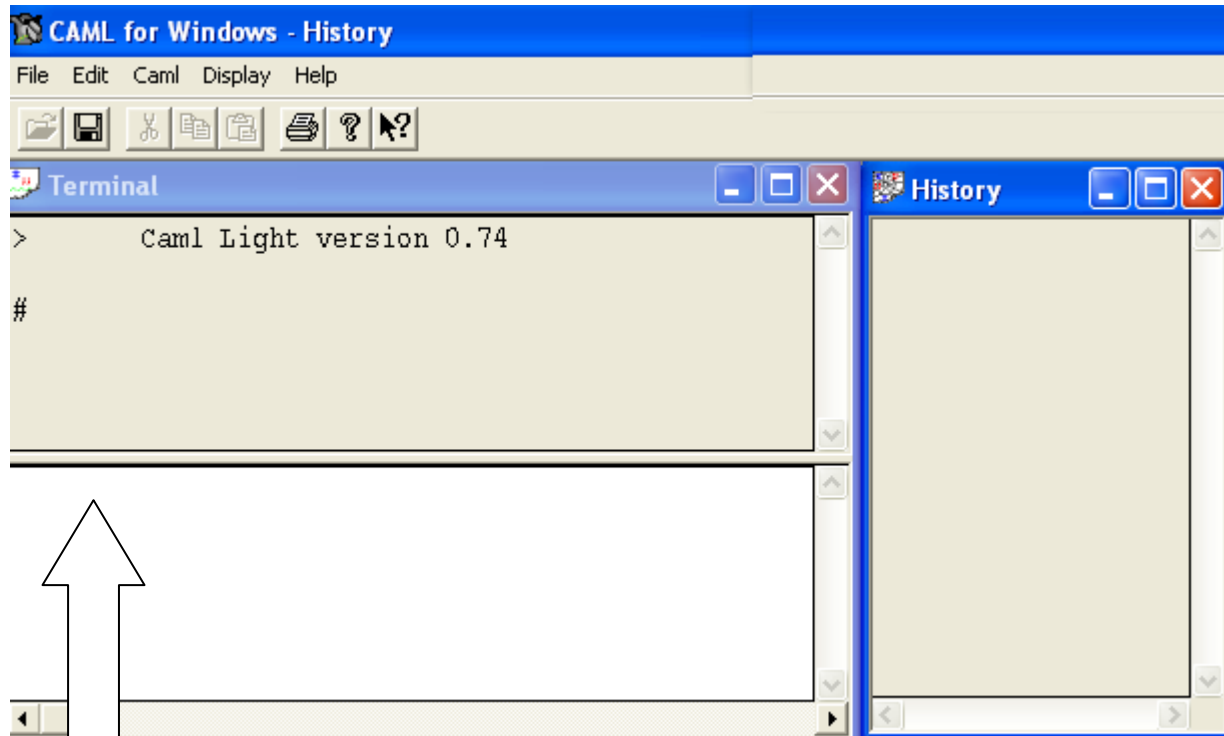


## 1. Lancement de CaML light

Ceci a pour effet de déclencher l'exécution de la boucle de **Lecture/Evaluation/Impression** qui se manifeste par les 2 fenêtres : Terminal (Editeur et Interprétation) et History (trace des expressions).



EDITEUR

En mode interactif, CaML donne systématiquement une réponse qui contient :

- Le nom de la variable déclarée s'il y en a.
- Le type trouvé pendant le typage.
- La valeur calculée après exécution.

## 2. Evaluation d'expressions

## TP 1 CaML Light

Pour vous entrainer à l'utilisation de CAML, demandez à l'interprète d'évaluer les expressions suivantes (et chercher à chaque fois à donner un sens à la réponse de l'interprète) :

### Expressions sur les entiers :

1098 ;;	5 * 99 ;;
10 / 5 ;;	10 / 6 ;;
276 mod 23 ;;	(3 * 5) + (10 - 6) ;;
21 + 35 + 12 + 7 ;;	2.7 + 10 ;;
int_of_float 1.1;;	int_of_float 1.1;;

### Expressions sur les réels (flottants) :

3.45;;	10.0 /. 6.0 ;;
1.1 +. 2.2 ;;	ceil 3.14 ;;
floor 3.14 ;;	exp 1.0 ;;
log 2.71828182846 ;;	3. ** 2. ;;
3.45e10;;	sqrt 25. ;;

### Expressions sur les booléens :

true ;;	not true ;;
true && false ;;	true or false ;;
1<2;;	1<=2;;
0<1 & 3<9;;	float_of_int 76;;
“un” < “deux”	(not false) or true ;;

### Expressions sur les caractères et chaînes de caractères :

`&` ;;	`a` ;;
int_of_char `a`;; (donne un type entier)	char_of_int 97;;
“a”;;	“Ceci est une ch” ^ “aine de caractères !”
“toutou”.[5] ;;	“kArim”.[1] ;;
string_length “KADER” ;;	sub_string “bonjour” 0 3

### Expressions sur les Uplets

(1,2) ;;	(“coucou”,3.1,(‘A’,2)) ;;
1,2,3 ;;	fst (12, “octobre”) ;;
(1,(2,3)) ;;	snd (12, “octobre”) ;;
(12, “octobre”) ;;	
(12, “octobre”,true) ;;	

## TP 1 CaML Light

### Définition :

Une définition est une déclaration qui associe un nom à une valeur. On distingue les déclarations globales les déclarations locales.

### Expressions de déclarations globales :

Syntaxe : **let** *nom* = *expr* ;;

où *nom* représente l'identificateur et *expr* l'expression qui lui est associée.

let x = 5 ;;	let pi = 3.14159 ;;
x ;;	let rayon = 10.0 ;;
x + 2 ;;	(pi *. rayon *. rayon) ;;
	let circonference = (2.0 *. pi *. rayon) ;;
let taille = 2.0 ;;	
taille ;;	
5 * taille ;;	

### Expressions de déclarations globales simultanées :

Syntaxe : **let** *nom*<sub>1</sub> = *expr*<sub>1</sub>

**and** *nom*<sub>2</sub> = *expr*<sub>2</sub>

.

.

**and** *nom*<sub>n</sub> = *expr*<sub>n</sub> ;;

let a= 3 and b= 2 ;;	
a + b ;;	

### Expressions de déclarations locales :

Syntaxe : **let** *nom* = *expr*<sub>1</sub> **in** *expr*<sub>2</sub>;;

la valeur *expr*<sub>1</sub> du nom *nom*, n'est connue que pour le calcul de *expr*<sub>2</sub> .

let x =2 in x * x ;;	let a = 1 and b = 2 in 2*a+b ;;
x ;;	let a = 1 and b = 2 * a in b + a;;
let y = 3 in x + y ;;	let a=1 in let b=2*a in b+a ;;
y;;	
let y = (let x = 3 in x + 3) ;;	
let x =3 in y = x + 3 ;;	
let x=(1 , "coucou") and y=("hello", 2.1) in (snd x , fst y) ;;	

### Expressions de conditions :

Syntaxe : **if** *expr*<sub>Bool</sub> **then** *expr*<sub>1</sub>**else** *expr*<sub>2</sub> ;;

La valeur de cette expression est la valeur de *expr*<sub>1</sub> si l'expression booléenne *expr*<sub>Bool</sub> s'évalue à **true** et la valeur de *expr*<sub>2</sub> sinon (à **false**).

if true then 1 else 0 ;;	let x=3 in if x=0 then 0 else 15/x ;;
(if 3=5 then 8 else 10) + 5 ;;	