Informatique DEUG A1 Partiel 13/4/94

Durée 2 heures. Aucun document.
A la fin de l'épreuve, déposer votre copie dans le paquet de votre groupe de td.

Question 1
a- Après évaluation de (define a 3), que donne l'évaluation de:
(let ((b (+ a 1)) (a (* 2 a))) (+ a b)) ?
b- Après évaluation de (define x '(3 2 (1 4 5))), que donne l'évaluation de:
(let ((z (caddr x))) (cons 1 (list (cadr x) (cons 3 (cdr z))))) ?
c- Combien d'éléments comporte la liste x ci-dessus ?

Question 2
On considère la fonction suivante:
```
(define (truc a b c d e f)
  (if (= a 0) (list a b c d e f) (truc b c d e f a)))
```
a- Evaluer:
(truc 2 1 0 8 9 3)
b- Que se passe-t-il si on évalue (truc 2 9 7 8 5 1) ?

Question 3
On considère les deux fonctions suivantes:
```
(define (toto x) (sqrt (+ 1 (* (cos x) (cos x)))))
(define (totobis x) (sqrt (+ 1 (* (log x) (log x)))))
```
Faire une fonction à deux arguments (super-toto x f) telle que l'on puisse redéfinir toto et totobis en posant:
```
(define (toto x) (super-toto x cos))
(define (totobis x) (super-toto x log))
```

Question 4
Faire une fonction (u n) calculant le terme u_n de la suite définie par:
$u_0 = 1$,
$$u_n = \frac{u_{n-1}}{1 + u_{n-1}}$$
(On demande une version *optimisée*).

Question 5
a- Faire une fonction (diviseur-suivant n d) calculant le plus petit diviseur de l'entier n qui soit supérieur ou égal à d.
b- En déduire une fonction (decompose n d) telle que l'appel (decompose n 2) calcule sous la forme d'une liste la décomposition en facteurs premiers de l'entier n, les facteurs premiers étant disposés en ordre croissant, et apparaissant autant de fois que leur exposant dans la factorisation de n. Exemple:
pour $60 = 2^2 * 3 * 5$: (decompose 60 2) -> (2 2 3 5),
pour $32 = 2^5$: (decompose 32 2) -> (2 2 2 2 2).

PARTIEL INFORMATIQUE DEUG A1 FORMULAIRE DE REPONSE 13/4/94

NOM :
Prénom :
Numéro de groupe de td : **Numéro d'appel :**

1a 10

1b (1 2 (3 4 5))

1c 3 éléments

2a (0 8 9 3 2 1)

2b Boucle infinie

3
```
(define (super-toto x f)
.       (sqrt (+ 1 (* (f x) (f x)))))
.
.
```

4
```
(define (u n)
.       (cond ((= n 0) 1)
.             (else (let ((x (u (- n 1))))
.                     (/ x (+ 1 x))))))
. Remarque: autre solution trouvée dans une copie d'étudiant...
. (define (u n) (/ 1 (+ 1 n)))
```

5a
```
(define (diviseur-suivant n d)
.     (cond ((= (remainder n d) 0) d)
.           (else (diviseur-suivant n (+ d 1)))))
.
.
.
```

5b
```
(define (decompose n d)
.     (cond ((= n 1) '())
.           (else (let ((d (diviseur-suivant n d)))
.                   (cons d (decompose (quotient n d) d))))))
.
.
.
```