Université de Perpignan — Année 2003-2004
Licence d'Informatique L2 — B. Goossens

Examen partiel de Programmation Impérative

Décembre 2003, durée 1h30, tous documents autorisés

Exercice 1 (éléments lexicaux). Parmi les propositions suivantes, quelles sont celles qui sont des identificateurs Ada ?

1. a:
2. http://www.univ-perp.fr
3. http
4. univ-perp

Parmi les propositions suivantes, quelles sont celles qui sont des littéraux numériques Ada ?

1. .
2. 0.
3. .0
4. 0.0
5. 00
6. 00.00
7. 0E0
8. 1E-1

Exercice 2 (affectation et test). Exprimer la condition inverse de :

```
(c>='A' and c<='Z') or (c>='a' and c<='z')
--vrai si c est une lettre majuscule ou minuscule
```

On rappelle que l'attribut `T'Val(x)` retourne la valeur d'énumération `T` de rang `x`. Par exemple, `Boolean'Val(1)` vaut `True`. `Character'Val(65)` vaut `'A'`. `Character'Val(48)` vaut `'0'`. Que calcule la fonction `E_vers_C` ? :

```
subtype Chiffre_Hexa is Integer range 0..15;
function E_vers_C(v: Chiffre_Hexa) return Character is
begin
  if (v<=9) then
     return (Character'Val(v+48));
  else
     return (Character'Val(v+55));
  end if;
end E_vers_C;
```

On rappelle que l'attribut `T'Pos(x)` retourne le rang de la valeur `x` d'énumération `T`. Par exemple, `Boolean'Pos(True)` vaut `1`; `Character'Pos('A')` vaut `65`. Que calcule `C_vers_E` ? :

```
subtype Chiffre_Hexa is Integer range 0..15;
function C_vers_E(h: Character) return Chiffre_Hexa is
  --precondition: h appartient a ['a'-'f']['A'-'F']['0'-'9']
  if (h>='a' and h<='f') then
     return (Character'Pos(h)-Character'Pos('a')+10);
  elsif (h>='A' and h<='Z') then
     return (Character'Pos(h)-Character'Pos('A')+10);
  else --(h>='0' and h<='9')
     return (Character'Pos(h)-Character'Pos('0'));
  end if;
```

Exercice 3 (fonctions). Écrire une fonction `Octet_vers_String` qui convertit un octet en une chaîne de deux caractères correspondant à sa représentation en base 16. Par exemple, `Octet_vers_String(108)` retourne "6C" (108 = 6*16 + 12). On rappelle que l'opérateur "/" appliqué à des arguments entiers retourne le quotient entier et l'opérateur "rem" fournit le reste (108/16=6, 108 rem 16=12). La fonction pourra faire appel à `E_vers_C`. Elle aura pour prototype:

```
subtype Octet is Positive range 0..255;
function Octet_vers_String(x: Octet) return String;
```

Écrire une fonction `String_vers_Octet` qui effectue la conversion inverse (String vers Octet). On supposera que la chaîne fournie en entrée se compose de deux caractères exactement, chacun représentant un chiffre hexadécimal. On rappelle que le premier caractère d'une chaîne a l'indice 1 (dans s="AB", s(1)='A' et s(2)='B').

```
subtype Octet is Positive range 0..255;
function String_vers_Octet(s: String) return Octet;
```

Exercice 4 (boucles). Soit la fonction suivante :

```
function f(x: Natural) return Natural is
  c, i, j: Positive:=1;
begin
  while (c<=x) loop
    j:= j+2;
    c:= c+j;
    i:= i+1;
  end loop;
  return (i-1);
end f;
```

Que vaut `f(10)` ? Et `f(16)` ? Quelle suite décrit la variable `i`? Même question pour la variable `j`. Même question pour la variable `c`. Que calcule la fonction `f` ?