

PCL - Exemples de programmes pour les évaluations

Ce document a pour but de fournir un guide dans votre progression de génération de code. On distinguera 4 “niveaux” dans les exemples, sachant qu’un niveau n’est qu’une indication générale sur votre progression. Bien que la moyenne au projet soit fixée sur une démonstration de “niveau 2”, la note finale dépend également des autres étapes et évaluations précédentes à celles de la production de code.

Niveau 1 : programme principal, variables, structures de contrôles élémentaires

Ce niveau comprend :

- l’exécution d’un programme principal,
- les déclarations de variables de types simples,
- les structures de contrôles : `if`, `for`,
- les affectations et expressions arithmétiques.

Un exemple de programme illustrant le niveau 1 est le suivant :

```
begin
  integer n, m, cpt, i;
  n := 100;
  m := 5;
  cpt := 0;

  for i := 0 step 1 until n do
    begin
      if (i % m = 0) then
        begin
          cpt := cpt + 1;
          outinteger(1, i)
        end
      end
    end;
    outinteger(1, cpt)
  end
```

Niveau 2 : Niveau 1 implémenté, appels de fonctions avec paramètres passés par valeur

Pour valider le *niveau 2*, il faut valider le *niveau 1* ainsi que :

- les définitions de fonctions,
- les appels de fonctions avec paramètres passés par valeurs, les retours de valeur.

Un exemple de programme illustrant ce niveau est le calcul de *factoriel(n)* itératif, avec appel de la fonction dans un programme principal.

Niveau 3 : Niveaux 1 et 2 implémentés, récursivité, gestion de la visibilité des blocs

Des exemples de code illustrant ces 2 aspects sont donnés dans les programmes *procedure11.a60* et *procedure9.a60* de *legopieces*.

Niveau 4 : Niveaux précédents implémentés, gestion des étiquettes, tableaux

Le fichier *vector7.a60* est un exemple de programme qui illustre ce dernier niveau, bien qu'on ne demande pas de gestion des réels dans ce niveau (adaptez cet exemple aux entiers uniquement).

Bonus :

Le passage des paramètres par nom, et la gestion des réels sont des aspects du langage à ne traiter que si toutes les autres structures du langage ont été implémentées.

Remarques :

On ne demande pas de gérer des entrées (lectures au clavier), ni des chaînes de caractères complexes. Dans premier temps, un affichage avec *outinteger()* suffira. L'affichage de chaînes de caractères simples pourra être un complément, de même que l'affichage des réels.