

Les 13 commandements pour la programmation élémentaire en Maple ¹

Ecole Polytechnique

Ne pas utiliser l'aspect réflexif du langage

- 1- ne pas utiliser les opérateurs qui gèlent l'exécution, comme les apostrophes dans 'expression' ou la fonction `evaln`.
- 2- ne pas utiliser les fonctions qui dégèlent l'évaluation d'une expression, comme `eval`.
- 3- ne pas utiliser de variables indéfinies. En Maple, les variables indéfinies ont leur nom comme propre valeur. Du coup on peut manipuler les programmes qu'on est en train d'écrire (réflexivité). C'est parait-il commode pour faire du calcul formel, mais ce n'est pas dans l'esprit de notre épreuve.

Ne pas utiliser les mémo-fonctions ou le hachage

- 4- ne pas utiliser l'option `remember` qui tabule les valeurs d'une fonction. On n'hésitera pas à recalculer plusieurs fois la valeur d'une fonction comme dans tout langage de programmation.
- 5- ne pas utiliser les tables, toujours préférer les tableaux. Les tables sont des tableaux associatifs avec hachage sur la clé de recherche. L'accès à un élément d'une table n'est pas une opération de coût unitaire.

Arguments/résultats de fonctions

- 6- ne pas hésiter à passer en argument les paramètres, entiers, flottants et tableaux, et à renvoyer les tableaux comme valeurs.
- 7- toujours déclarer les variables scalaires ou les tableaux, avec leur qualificatif local ou global. Rappeler qu'on est forcé de mettre global à cause des déclarations implicites de variables comme locales.
- 8- ne pas utiliser les fonctions comme objets de première classe.
- 9- ne pas emboîter les définitions de fonctions.

Types

- 10- ne pas faire de test dynamique sur les types. On doit s'aligner sur les autres langages de programmation qui ont plutôt tendance à faire du typage statique à la compilation.
- 11- utiliser les types standard (integer, float). Typer autant que possible les arguments et le résultat d'une fonction, même si certaines versions de Maple n'en tiennent pas compte ou pire ne les accepte pas.
- 12- utiliser les scalaires comme s'ils étaient de taille fixe (ie sur 32 bits), et ainsi supposer que le coût d'une opération arithmétique est unitaire.

Graphique

- 13- pour faire l'affichage graphique d'un tableau de points, on utilisera la fonction `afficherPoints(t)`

```
afficherPoints := proc (a)
    plot (convert (a, list));
end;
```

ou, mieux, on peut utiliser le paquetage `quickdraw`.

Remarque générale

Il n'y a pas besoin de connaître les fonctions de la bibliothèque standard Maple, hors les fonctions d'impression pour mettre au point les programmes en séances de TD. Pour l'épreuve du concours de l'X, si on veut utiliser une fonction de librairie (graphique ou numérique), la fonction sera toujours donnée avec sa signature et sa sémantique.

¹Pour la petite épreuve d'informatique du concours d'admission de l'Ecole Polytechnique, ces commandements ne sont nullement impératifs. Ce sont des recommandations pour rester dans l'esprit de cette épreuve.