

Pretty printer

Concevoir un programme permettant de justifier un texte. Le texte sera lu depuis un fichier (on pourra utiliser `istream::getline()` pour extraire une ligne du fichier) et la sortie devra être de sorte que le texte sera calé à gauche, à droite, centré ou justifié de sorte qu'aucun mot ne soit coupé par un retour à la ligne, ni aucune phrase (suite de mots terminée par un point ou exclamation, etc). La largeur du texte final devra pouvoir être choisie par l'utilisateur. Pour que n mots (y compris la ponctuation) tiennent sur une ligne, il doit y avoir au moins la place pour $n - 1$ espacements mais qu'en ajoutant le mot suivant la place soit insuffisante. S'il reste plus d'espacement que nécessaire on les distribuera aléatoirement (ou pas mais c'est plus «joli» avec) de sorte à «comblent» la ligne. Attention, dans le cas de la justification aucune ligne (sauf la dernière d'un paragraphe) ne doit commencer ni finir par un espacement. On se contentera de découper simplement le texte d'origine en mots : un mot est une suite de lettres sans caractère d'espacement ni ponctuation. On notera que certains mots (les mots de la fin d'une phrase) sont suivis d'une ponctuation.

La conception pourrait avantageusement se structurer ainsi: On introduit le concept de Boîte qui représente les objets composant le texte formaté.

- une Boîte est un élément du texte formaté qui a une taille et peut être affiché.
- une BoîteEspace est un élément du texte formaté qui séparera les mots. Ce sera un seul espace dans le cas où le texte n'est pas justifié, et potentiellement plusieurs espaces sinon.
- une BoîteMot est un élément du texte formaté qui représentera un mot.
- une BoîteComposite représentera une ligne de texte et contiendra d'autres boîtes
- le Formateur qui utilisera le Parseur et les objets précédemment décrits pour construire le texte formaté.