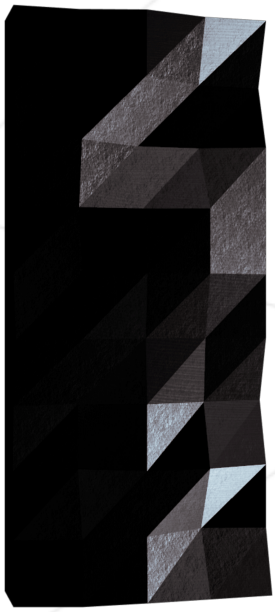

Piscine C

Rush 02

Résumé: Ce document est le sujet du rush 02 de la piscine C de 42.

Table des matières

Chapitre I

Consignes

- Le groupe sera automatiquement inscrit en soutenance.
- Toute demande de précision sur le sujet complexifiera probablement le sujet.
- Vous devez suivre la procédure de rendu pour tous vos exercices.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- La Moulinette compile avec les flags -Wall -Wextra -Werror, et utilise `cc`.
- Si votre programme ne compile pas, vous aurez `0`.
- Vous devez rendre un Makefile, qui compile votre projet à l'aide des règles `$NAME` `clean` et `fclean`
- Vous devrez donc réaliser le sujet indiqué avec les binômes imposés et vous présenter en soutenance à l'heure dite avec tous vos binômes.
- Lors de la soutenance, le projet devra être terminé. Les soutenances servent à presenter et à expliquer votre travail dans les moindres détails.
- Chaque membre du groupe devra parfaitement être au courant du travail réalisé, chacun des membres sera interrogé, la note du groupe étant basée sur les moins bonnes explications.
- évidemment, vous devez tout faire pour prendre contact avec vos binômes : téléphone, mail, pigeon voyageur, séance de spiritisme, etc. Aucune excuse ne sera acceptée en ce qui concerne les problèmes de groupe.
- Si après avoir vraiment tout essayé un de vos binômes reste injoignable : réalisez votre rush on s'arrangera en soutenance. Même si c'est le chef de groupe : vous avez tous accès au dépôt.
- Bien sûr, votre travail devra être à la Norme : soyez très rigoureux. Bon travail !

Chapitre II

Préambule

Voici la recette du Quatre-Quart :

```
4 Ingrédients (Pour 1 gateau) :
- 4 oeufs
- 200g de sucre
- 200g de farine
- 200g de beurre demi-sel

Préparation :
- Préchauffez le four à 180°C.
- Commencez par faire fondre le beurre pour lui laisser le temps de refroidir.
- Fouettez les œufs avec le sucre pendant 5 minutes environ.
Il faut que le mélange devienne clair et mousseux.
- Ajoutez la farine (et éventuellement l'arôme de votre choix) et
fouettez quelques secondes, juste le temps de l'incorporer.
Si vous fouettez trop, vous allez faire retomber toute votre pâte.
- Ajoutez le beurre fondu et mélangez avec une maryse
ou une cuillère en bois.
- Beurrez votre moule à cake et versez la préparation à l'intérieur.
- Plantez la lame d'un couteau dans la pâte dans toute la longueur du cake.
- Enfournez 10 minutes à 180°C, puis baissez la température du four à 145°C.
- Laisser cuire encore 40 minutes.

Le dessus du quatre-quarts doit être légèrement doré, mais pas trop.
```

FIGURE II.1 – Ca devrait ressembler à ça.

Chapitre III

Le sujet

	Exercice : 00
rush-02	
Dossier de rendu : $ex00/$	
Fichiers à rendre : `Makefile et tous les fichiers nécessaires`	
Fonctions Autorisées : `write, malloc, free, open, read, close`	

- Vous devez réaliser un programme qui prend un nombre en argument en entrée et qui le converti en sa valeur écrite.
- Nom de l'exécutable : `rush-02`
- Votre code source sera compilé par la commande :

```
make fclean
make
```

- Votre programme peut recevoir jusqu'à 2 arguments :
  - Si il n'y a qu'un seul argument, il s'agit de la valeur que vous devez convertir.
  - Si il y a 2 arguments, le premier est le dictionnaire à utiliser, et le deuxième est la valeur à convertir.
- Si l'argument n'est pas un entier positif valide, vous devrez renvoyer "Error" suivi d'un "\n".
- Si le dictionnaire ne vous permet pas de convertir la valeur demandée, vous devrez renvoyer "Error" suivi d'un "
n"
- Pour des soucis d'harmonisation, votre programme parlera anglais.
- Votre programme doit parser le dictionnaire passé en ressources. Les valeurs données doivent être utilisées pour imprimer les résultats. Celles-ci pourront être mo-

difiées.

- Toute mémoire allouée sur la heap (avec malloc(3)) doit être libérée proprement.
- Le dictionnaire suivra les règles suivantes :

```
[a number][0 to n spaces]:[0 to n spaces][n'importe quel caractere imprimable]\n
```

  - Vous devez trimmer les espaces en début et en fin de chaine de caractères.
  - Le dictionnaire aura toujours au moins les clés données dans le dictionnaire de réference. Leur valeur peut être modifiée, des entrées peuvent êre rajoutées, mais les clés initiales ne peuvent pas être retirées.
  - Vous ne devez utiliser que les valeurs initialement données dans le dictionnaire en annexe. (Par exemple, si l'on rajoute la clé "54 : fifty four", vous devez tout de même utiliser les clés "50 : fifty et 4 : four")
  - Les entrées du dictionnaire peuvent être rangées dans n'importe quel ordre.
  - Il peut y avoir des lignes vides dans le dictionnaire.
  - Si vous avez la moindre erreur lors du parsing du dictionnaire, vous devez afficher "Dict Error\n". Votre programme doit quitter ensuite proprement.
  - Si le dictionnaire ou votre programme ne vous permet pas de résoudre la valeur demandée, vous afficherez "Dict Error\n".
- Example :

```
$> ./rush-02 42 | cat -e
forty two$
$> ./rush-02 0 | cat -e
zero$
$> ./rush-02 10.4 | cat -e
error$
$> ./rush-02 100000 | cat -e
one hundred thousand$
$> grep "20" numbers.dict | cat -e
20   :     hey     everybody !$
$> ./rush-02 20 | cat -e
hey everybody !$
```

Chapitre IV

Bonus

- Utiliser des clés-valeurs custom : `54: fifty-four`, au lieu de `50: fifty` and `4: four`
- Utiliser `,`, `-`, `and` pour être syntaxiquement correct.
- Avoir la possibilité de changer de langage (ex. le faire en français). Vous pouvez à cet effet rajouter votre propre dictionnaire qui contiendra les entrées nécessaires.sa
- Utiliser `read` pour lire l'entrée standard lorsqu'il n'y a pas d'argument
- Si vous avez d'autres idées et qu'elles apportent quelque chose au projet, allez-y !