# Projet final

mardi 29 mars 2022 10:02

## Introduction

Le projet porte sur la recherche de motifs (séquence de lettres correspondant à une expression régulière) dans une chaîne de caractères formée à partir d'un alphabet de 4 lettres. L'alphabet est constitué des 3 premières consonnes et de la première voyelle du nom de famille de l'étudiant. Si nécessaire, une ou plusieurs consonnes du prénom peuvent être utilisées.

Exemples :

| | |
|---|---|
| Balmas | blma |
| Dupont | dptu |
| Roy Thomas | rtho |
| Meyer Pierre | mrpe |

Les motifs sont formés avec les 3 caractères spéciaux suivants :

- * le caractère précédent apparaît de 0 à n fois
- + le caractère précédent apparaît de 1 à n fois
- ? le caractère précédent apparaît de 0 ou 1 fois

Exemples de motifs :

- xy*z x, suivi de 0 à n y, suivi de z soit xz, xyz, xyyz, xyyyz, xyyyyz, etc
- wxz+ w, suivi de x, suivi de 1 à n z soit wxz, wxzz, wxzzz, wxzzzz, etc
- wx?y w, suivi 0 ou 1 x, suivi de y soit wy et wxy
- wx?y*z wz, wyz, wxz, wxyz, wyyz, wxyyz, par exemple
- wx+y*z wxz, wxyz, wxxyz, wxyyz, par exemple

Chaque étudiant doit envoyer un mail à l'enseignante avec son nom et son prénom, ainsi que son alphabet personnel. Il recevra en retour les 2 motifs qu'il utilisera dans son projet. Les différentes étapes du projet sont décrites ci-dessous.
Toutes les étapes de programmation doivent être réalisées en langage C.

## Etape 1

Donnez le schéma de l'automate correspondant à chacun des 2 motifs, ainsi que les tables de transition correspondantes.

## Etape 2

Remplissez un tableau de 200 caractères avec les lettres de votre alphabet personnel, et uniquement celles-ci.
Recherchez toutes les occurrences du premier motif et constituez l'ensemble des occurrences trouvées avec leur position (indice) dans le tableau + longueur de l'occurrence.
Calculez également le nombre total d'occurrences trouvées.
Affichez ces informations.

## Etape 3

Faites de même avec le 2ème motif.

## Etape 4

Constituez un ensemble avec toutes les occurrences des 2 motifs trouvées mais sans doublons.
Affichez cet ensemble.

## Etape 5

Classez ces occurrences dans l'ordre croissant de leur longueur.
Affichez le résultat.

## Etape 6

Vérifiez que les algorithmes utilisés pour la recherche des 2 motifs (étape 2 et 3) sont corrects.
Pour cela vous devez utiliser un tableau non aléatoire et proposer une vérification automatique (test inclus dans le programme).

## Etape 7

Refaites les étapes 2 à 5 avec, successivement, un tableau de 2'000, 20'000, 200'000 lettres de votre alphabet.
Pour chaque taille, vous afficherez uniquement le nombre d'occurrence pour chaque motif, ainsi que le nombre de motifs sans doublons.

## Etape 8

Réfléchissez à la complexité des algorithmes que vous avez implémentés pour la recherche des 2 motifs (étapes 2 et 3). Comment se comportent-ils quand la taille du tableau passe de 200 à 2'000, puis 20'000 et 200'000 ? Comptez les tours de boucles, ou testez les temps d'exécution, et analysez … Analysez en particulier l'évolution des tours de boucle / temps d'exécution par rapport à l'évolution de la taille du tableau.
Pensez-vous avoir trouvé l'algorithme (presque) optimal ? Si oui, justifiez en détail votre réponse. Si non, expliquez pourquoi et dites comment vous auriez pu faire pour mieux faire.

## Etape 9

Intégrez le programme ci-dessous (avec toutes les adaptations nécessaires) à votre propre programme. Faites-le tourner pour le tableau de 200 lettres.

```
/*
Tableau des lettres :
"xyzzwyxxzyyyxywyzxyxzzzyzyzxxwyyyxyyzzzyxywzwxxzyzwxyzzxxzzzzxxyyzw
xzwwwywzyxwyzzywxywzzzyyyzwwxzxyyxyzzzyxwzzwyywwwwzzzxyywyyxxzzwwyzz
xxxxwwxzyyzyyyy"

Tableau des resultats :
4, 5, 44, 4, 66, 3, 94, 3, 123, 6, 140, 3
*/

void affiche_motifs(int *taboccur, int indocc, int nbocc, char *str)
{
  if (indocc >= nbocc*2) {
    return;
  }
  affiche_un_motif(str, taboccur[indocc],
taboccur[indocc]+taboccur[indocc+1]);
  affiche_motifs(taboccur, indocc+2, nbocc, str);
}

void affiche_un_motif (char *str, int deb, int fin) {
  if (deb>=fin) {
    printf("\n");
    return;
  }
  printf("%c", str[deb]);
  affiche_un_motif (str, deb+1, fin);
}
```

```
/*
Appel : affiche_motifs avec 4 arguments
 1. tableau des resultats
 2. 0 (valeur d'initialisation)
 3. nombre d'occurrences du motif
 4. tableau des lettres

Ex : affiche_motifs (tabresultat, 0, 6, tablettres);
*/
```

## Etape 10

Transformez les 2 fonctions récursives en fonctions itératives. Faites-les tourner pour le tableau de 200 lettres.

## Etape 11

Pensez-vous que vous pouvez faire tourner les 2 fonctions récursives sur les tableaux de 2'000, 20'000, 200'000 lettres voire plus ? Argumentez votre réponse de manière détaillée.

## A remettre sur le Moodle pour le mardi 3 mai fin de soirée (délai strict)

Une archive zip ou tgz (à l'exclusion de tout autre format) contenant :

- Code source de toutes les étapes (sans exécutable)
- Traces d'exécution des étapes 2 à 5 ainsi que 7
- Traces d'exécution du programme de test (étape 6)
- Traces d'exécution des fonctions récursives et itératives (étapes 9 et 10)
- Explications sur le programme de test (étape 6)
- Rédaction sur la question complexité (étape 8)
- Rédaction sur la question récursion (étape 11)

Vous rassemblerez traces d'exécution et rédactions dans un unique document PDF.

**Attention** :

- Le travail est à faire individuellement, toute copie sera sanctionnée
- La qualité de présentation sera prise en compte
- La qualité du français (orthographe et grammaire) sera prise en compte
- Tout retard de remise sera pénalisé

## Examen oral mardi 10 et mercredi 11 mai (13h30-16h30)

Vous devrez :

- Présenter (démo) votre projet
- Répondre à des questions

# Exemple pour les étapes 2 à 5

mardi 29 mars 2022 10:25

Alphabet : wxyz

Tableau :
xyzzwyxxzyyyxywyzxyxzzzyzyzxxwyyyxyyzzzyxywzwxxzyzwxyzzxxzzzzxxyyzwxzwwwywzyxwyzzywxywzzzy
yyzwwxzxyyxyzzzyxwzzwyywwwwwzzzxyywyyxxzzwwyzzxxxxwwxzyyzyyyy

motif 1 à chercher : wy*x+z

6 occurrences trouvées

| | | |
|---|---|---|
| wyxxz | 4 | 5 |
| wxxz | 44 | 4 |
| wxz | 66 | 3 |
| wxz | 94 | 3 |
| wyyxxz | 123 | 6 |
| wxz | 140 | 3 |

6 occurrences différentes des 2 motifs trouvées

| | |
|---|---|
| wxz | 3 |
| wxxz | 4 |
| wyyxxz | 6 |
| wz | 2 |
| wyz | 3 |
| wyxxz | 5 |

motif 2 à chercher : wy?x*z

13 occurrences trouvées

| | | |
|---|---|---|
| wyxxz | 4 | 5 |
| wyz | 14 | 3 |
| wz | 42 | 2 |
| wxxz | 44 | 4 |
| wxz | 66 | 3 |
| wz | 73 | 2 |
| wyz | 77 | 3 |
| wz | 85 | 2 |
| wxz | 94 | 3 |
| wz | 107 | 2 |
| wz | 116 | 2 |
| wyz | 131 | 3 |
| wxz | 140 | 3 |

Les mêmes triées

| | |
|---|---|
| wz | 2 |
| wyz | 3 |
| wxz | 3 |
| wxxz | 4 |
| wyxxz | 5 |
| wyyxxz | 6 |