

Projet final 2023

lundi 13 mars 2023 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, xyyyyz, xyyyyyz, 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 indice de début et indice de fin dans le tableau.

Calculez également le nombre total d'occurrences trouvées.

Affichez ces informations.

Etape 3

Faites de même avec le 2ème motif.

Etape 4

Constituez l'union de toutes les occurrences des 2 motifs trouvées, mais sans doublons.
Affichez cet ensemble.

Etape 5

Classez ces occurrences dans l'ordre décroissant de leur nombre d'occurrences.
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

Vérifiez que l'algorithme utilisé pour le calcul de l'union (étape 4) est correct (vérification automatique)

Etape 8

Vérifiez que l'algorithme utilisé pour le classement (étape 5) est correct (vérification automatique)

Etape 9

Intégrez le programme ci-dessous (avec toutes les adaptations nécessaires) à votre propre programme et faites-le tourner.

```
/*
Tableau des lettres :
"wwzwxyzxxzxyyyxxxxzyzxxzywxxyxyxxzyyxxwxxxyzxxwyyzwyxwwxxyyzxxwxy
zxwzyxwwwxxwzywwwxxyxzwxxywzxyzxwxxwxywyxzyxxzxxwxxwxyyxyzxxwzy
yzyzxxzzyzwxxyyzyxyxxzyzzxyyzxxxxxzwyyyxwxyywwzxywywxxwzywywxxzx"
```

```
Tableau des resultats (occurrences) :
1 2   4 7   48 51   65 67   69 70   79 80   96 97   118 120   132
133   145 150   180 181   189 191   195 197   4 7   40 44   118 120
189 191   195 197
```

Appel : affiche_motifs avec 4 arguments

1. tableau des resultats
2. 0 (valeur d'initialisation)
3. nombre d'occurrences des 2 motifs
4. tableau des lettres

```
Ex : affiche_motifs (tabresultat, 0, 18, tablettres);
*/
```

```
void affiche_motifs(int *taboccur, int indocc, int nbocc, char *str)
{
    if (indocc >= nbocc*2) {
        printf("\n");
        return;
    }
    affiche_un_motif(str, 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("    ");
        return;
    }
    printf("%c", str[deb]);
    affiche_un_motif (str, deb+1, fin);
}

```

Etape 10

Faites tourner votre programme, sans puis avec les fonctions ci-dessus intégrées, sur un tableau de lettres de successivement 2'000, 20'000, 200'000, 2'000'000, 20'000'000, 200'000'000, etc. Jusqu'à quelle taille pouvez-vous faire tourner le programme ? Expliquez la raison de cette limite.

Etape 11

Modifier le programme pour qu'il puisse tourner même sur des tableaux beaucoup plus grands que la limite rencontrée. Expliquez pourquoi cette version marche mieux pour des tableaux de très grande taille.

A remettre sur le Moodle pour le *date précise à déterminer (fin avril)* (délai strict)

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

- Schémas et tables de transitions de l'étape 1
- Code source de toutes les étapes (sans exécutable ni .o)
- Traces d'exécution des étapes 2 à 5 ainsi que 7 et 9
- Traces d'exécution des programmes de test des étapes 6, 7 et 8
- Explications sur les programmes de test des étapes 6, 7 et 8
- Rédaction sur la question de l'étape 10
- Code source + explications sur la question de l'étape 11

Vous rassemblez 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 *date précise à déterminer (début mai)*

Vous devrez :

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

Exemple 2023

jeudi 9 mars 2023 10:22

chaîne :

wwzwwxyzxxzxyyyxyxxzyzxxzywxxyxyyyxzyxxwxxxyzzxwwyyzwyxwwwxyyzxwxwyzxwzyxwwwxwwzywwwxxyxzw
xwxywzxyzxwwxwxywyxzyxzzxwxzwxwyyxyzzxwzyzyzxxzxyzwxxyyzxyxxyzzxxyyzxxxxxzwyyyxwxyywwzxy
wywwwxxyzywywxzxx

Motif 1 : wx?y*z

Affichage table des occurrences

wz	[1	2]
wxyz	[4	7]
wyyz	[48	51]
wyz	[65	67]
wz	[69	70]
wz	[79	80]
wz	[96	97]
wxz	[118	120]
wz	[132	133]
wxyyyz	[145	150]
wz	[180	181]
wxz	[189	191]
wxz	[195	197]

13 motifs 1 trouves

Motif 2 : wx+y?z

Affichage table des occurrences

wxyz	[4	7]
wxxyz	[40	44]
wxz	[118	120]
wxz	[189	191]
wxz	[195	197]

5 motifs 2 trouves

Union des motifs trouves sans doublons

wxxyz 1
wxyz 2
wyyz 1
wxz 6
wyz 1
wz 6
wxyyyz 1
--> 7 motifs uniques

Motifs tries

wxz 6
wz 6
wxyz 2
wxxyz 1
wyz 1
wyyz 1
wxyyyz 1

affiche_motifs :

wz	wxyz	wyyz	wyz	wz	wz	wz	wxz	wz	wxyyyz	wz	wxz	wxz	wxyz	wxxyz
wxz	wxz	wxz												