

Exercice 1 :

1) L'environnement d'un compilateur inclut plusieurs composants...

Exemple en C (ex. un programme "hello world"):

```
#include <stdio.h>

int main() {
    printf("Hello");
    return 0;
}
```

- Code source (.c) → Préprocesseur : Inclut `stdio.h`, supprime commentaires → Code prétraité (.i ou directement passé).
- Code prétraité → Compilateur : Analyse et traduction → Fichier objet (.o) en assembleur ou code machine.
- Fichier objet + bibliothèques → Éditeur de liens : Résout `printf()` de stdio → Exécutable relogeable (.exe).
- Exécutable → Chargeur : Charge en mémoire → Code exécutable absolu, prêt à exécuter.

2) Pour la structure conditionnelle :

- Code intermédiaire à trois adresses avec labels :

```
t1 = x > y
if t1 goto L1
t2 = y - x
result = t2
goto L2
L1: t3 = x - y
result = t3
L2: return result
```

Ce code utilise des temporaires (t1, t2, t3) pour les opérations, et des labels (L1, L2) pour les sauts conditionnels et inconditionnels.

- Optimisation : On reconnaît que cette structure calcule la valeur absolue $|x - y|$. Code optimisé :

```
t1 = x - y
if t1 < 0 goto L1
result = t1
goto L2
```

```

L1: result = -t1
L2: return result

```

Ou plus simplement (en simplifiant les calculs) :

```

t1 = x - y
result = abs(t1)
return result

```

3) Pour la boucle simple :

a) Code intermédiaire à trois adresses avec labels :

```

i = 0
L1: t1 = i < 5
if not t1 goto L2
t2 = i * 2
sum = sum + t2
t3 = i + 1
i = t3
goto L1
L2: // fin de boucle

```

b) Optimisation (déroulement de boucle, car 5 itérations est petit) : Code optimisé (déroulé) :

```

sum = sum + 0 * 2 // i=0 → sum += 0
sum = sum + 1 * 2 // i=1 → sum += 2
sum = sum + 2 * 2 // i=2 → sum += 4
sum = sum + 3 * 2 // i=3 → sum += 6
sum = sum + 4 * 2 // i=4 → sum += 8

```

Ou simplifié davantage (propagation de constantes) :

```

sum = sum + 0 + 2 + 4 + 6 + 8
sum = sum + 20

```

Exercice 2 : Théorie des langages

Soit $\Sigma = \{0, 1, 2\}$.

1) Les mots de longueur 2 sont : 00, 01, 02, 10, 11, 12, 20, 21, 22. Il y en a 9. Calcul : Pour un alphabet de taille 3, le nombre de mots de longueur $n=2$ est $3^2 = 9$ (chaque position a 3 choix indépendants).

- 2) $w^2 = "012012"$ (concaténation de w avec lui-même). $w^3 = "012012012"$. Longueur de w^4 : $|w| = 3$, donc $|w^4| = 4 * 3 = 12$.
- 3) Un exemple de v : $"012"$ (car $"012"^3 = "012012012"$). Il est unique car la longueur de $"012012012"$ est 9, divisible par 3, et la répétition est exacte sans sous-périodes plus petites évidentes.
- 4) Σ^* est l'ensemble de tous les mots finis sur Σ , incluant le mot vide ϵ . Σ^+ est l'ensemble de tous les mots non vides sur Σ . Exemple : ϵ appartient à Σ^* mais pas à Σ^+ .

Exercice 3 : Théorie des langages

Soit $\Sigma = \{a, b\}$, $u = "aba"$, $v = "bb"$.

- 1) $u.v = "ababb"$; $v.u = "bbaba"$. Non commutative car $u.v \neq v.u$ ($ababb \neq bbaba$).
- 2) $w = u.v = "ababb"$. Préfixes propres : $a, ab, aba, abab$. Suffixes propres : $b, bb, abb, babb$. (Propres : non vides et non égaux à w).
- 3) $"ab"$ est un préfixe de w car w commence par $"ab"$. $"bb"$ est un suffixe de w car w finit par $"bb"$. Justification : Un préfixe est une sous-chaîne initiale, un suffixe une sous-chaîne finale.
- 4) $x.w = \epsilon."ababb" = "ababb"$; $w.x = "ababb".\epsilon = "ababb"$. Donc $x=\epsilon$ est neutre (ne change pas le mot concaténé).

Exercice 4 : Théorie des langages

Soit $\Sigma = \{x, y\}$, $w = "xyy"$.

- 1) $w^R = "yyx"$ (inversion). w n'est pas un palindrome car $"xyy" \neq "yyx"$.
- 2) Exemples : xxx, xyx, yxy (palindromes : égaux à leur miroir).
- 3) Ordre hiérarchique (longueur puis lexico, $x < y$) : ϵ (long. 0), x, y (long. 1, $x < y$), xy, yy (long. 2, $xy < yy$), xyy (long. 3).