

IFT209 – Programmation système
Université de Sherbrooke

Devoir 4

Enseignant: Michael Blondin
 Date de remise: mercredi 27 mars 2019 à 23:59
 À réaliser: en équipe de deux
 Modalités: remettre en ligne sur **Turnin**; une seule remise avec vos noms/CIP en commentaires en en-tête du code

Problème. Dans ce devoir, vous devez implémenter quelques manipulations de chaînes de caractères. Lorsque votre programme est exécuté, il doit lire une chaîne de caractères s au clavier, suivi d'un entier n compris entre 0 et 5. L'opération n est ensuite effectuée sur s , puis le programme se termine. Vous devez implémenter les opérations suivantes:

Taille d'une chaîne	
Opération	0
Entrée	Chaîne de caractères UTF-8 s
Effet	Affiche le nombre de caractères (non nuls) de s
Tests	abcdef → 6 é → 1 aéケヱ → 4

Casses et substitutions										
Opération	1									
Entrée	Chaîne de caractères ASCII s									
Effet	Affiche la chaîne obtenue en appliquant ces opérations à s : <ul style="list-style-type: none"> • Les lettres aux positions paires sont mises en minuscule et les lettres aux positions impaires sont mises en majuscule (la première position est 0); • Ces voyelles doivent être remplacées par ces chiffres: <table style="margin-left: auto; margin-right: auto; border: none;"> <tr> <td style="padding: 0 10px;">A ↦ 4</td> <td style="padding: 0 10px;">a ↦ 4</td> </tr> <tr> <td style="padding: 0 10px;">E ↦ 3</td> <td style="padding: 0 10px;">e ↦ 3</td> </tr> <tr> <td style="padding: 0 10px;">I ↦ 1</td> <td style="padding: 0 10px;">i ↦ 1</td> </tr> <tr> <td style="padding: 0 10px;">O ↦ 0</td> <td style="padding: 0 10px;">o ↦ 0</td> </tr> </table> • Les autres caractères sont inchangés. 	A ↦ 4	a ↦ 4	E ↦ 3	e ↦ 3	I ↦ 1	i ↦ 1	O ↦ 0	o ↦ 0	
A ↦ 4	a ↦ 4									
E ↦ 3	e ↦ 3									
I ↦ 1	i ↦ 1									
O ↦ 0	o ↦ 0									
Tests	<table style="margin-left: auto; margin-right: auto; border: none;"> <tr> <td style="padding: 0 10px;">brUN</td> <td style="padding: 0 10px;">→</td> <td style="padding: 0 10px;">bRuN</td> </tr> <tr> <td style="padding: 0 10px;">bonjour !</td> <td style="padding: 0 10px;">→</td> <td style="padding: 0 10px;">b0nJ0Ur !</td> </tr> <tr> <td style="padding: 0 10px;">Ceci est une phrase formidable</td> <td style="padding: 0 10px;">→</td> <td style="padding: 0 10px;">c3c1 3sT Un3 PhR4S3 f0rM1D4BI3</td> </tr> </table>	brUN	→	bRuN	bonjour !	→	b0nJ0Ur !	Ceci est une phrase formidable	→	c3c1 3sT Un3 PhR4S3 f0rM1D4BI3
brUN	→	bRuN								
bonjour !	→	b0nJ0Ur !								
Ceci est une phrase formidable	→	c3c1 3sT Un3 PhR4S3 f0rM1D4BI3								

Hexadécimal vers décimal	
Opération	2
Entrée	Chaîne de caractères ASCII s représentant une valeur hexadécimale non signée (comprise entre 0 et $2^{64} - 1$)
Effet	Affiche la valeur décimale de s
Tests	5 → 5 0A → 10 FF → 255 ABCDEF98 → 2882400152

Binaire vers décimal	
Opération	3
Entrée	Chaîne de caractères ASCII s représentant un entier signé sous représentation binaire (compris entre -2^{63} et $2^{63} - 1$)
Effet	Affiche la valeur décimale de s
Tests	0101 → 5 00111 → 7 10100 → -12 111111 → -1

Chiffrement par décalage	
Opération	4
Entrée	Chaîne de caractères ASCII s dont les caractères sont parmi: A, B, C, D, E, F, H, I, J, K, L, M, N, P, Q, R, S, T, U, V, X, Y, Z, [, \ et]
Effet	<p>La chaîne s a été chiffrée à partir d'une chaîne t. Vous devez afficher la chaîne t; autrement dit, vous devez <i>déchiffrer</i> s.</p> <p>La chaîne t est constituée exclusivement de lettres majuscules (A à Z). La chaîne s a été obtenue à partir de t en appliquant les deux transformations consécutives suivantes à chacune des lettres de t:</p> <p>(i) La lettre est décalée circulairement de 7 positions vers l'avant dans l'alphabet:</p> <p style="text-align: center;"> A ↦ H B ↦ I ⋮ ↦ ⋮ Y ↦ F Z ↦ G </p> <p>(ii) Les 5 bits de poids faible du caractère obtenu à l'étape (i) sont décalés circulairement de 3 bits vers la gauche.</p> <p>Exemple de la transformation:</p> <p>(i) C ↦ J (ii) J = 01001010₂ ↦ 01010010₂ = R</p>
Tests	R → C RRR → CCC HU]VCRNCH → TOPSECRET

Permutations	
Opération	5
Entrée	Chaîne de caractères ASCII s sans répétition de caractères
Effet	Affiche toutes les permutations de s (dans l'ordre de votre choix)
Tests	ab → ab ba abc → abc acb bac bca cba cab

Directives.

- Votre programme doit être obtenu en complétant le code partiel ci-dessous;
- Votre programme doit être remis dans un seul fichier nommé `devoir4.s`;
- Ne modifiez pas le point d'entrée ainsi que le format de la chaîne en entrée;
- Vous devez supposer que la chaîne s respecte le format de l'opération choisie (aucune validation);
- Vous ne pouvez pas utiliser `printf`, `scanf` ou d'autres fonctions d'une librairie existante afin d'effectuer les conversions.

Pointage. Vous pouvez obtenir un maximum de 20 points. Vous obtenez:

- 1 point si votre programme lit une chaîne de caractères et un code d'opération, et affiche un résultat;
- 1 point pour chaque opération qui passe les tests donnés plus haut (donc 6 points au maximum);
- 1,5 points par opération bien implémentée (donc 9 points au maximum);
- 2 points si votre code est bien indenté (codes d'opération, opérandes et commentaires alignés);
- 2 points pour la qualité et lisibilité du code (commentaires significatifs, organisation du code, pas de « code spaghetti », etc.)

Bonus. Vous obtenez 0,5 point bonus (absolu) à votre note finale de la session pour chacune de ces fonctionnalités additionnelles:

- l'opération 3 est implémentée sous forme de sous-programme avec *au plus dix* instructions (excluant les étiquettes) dont *au plus une* occurrence de `cbz`, *au plus une* occurrence de `b`, et *aucune* autre instruction de branchement (aucun `b.cond`, `bl`, `blr`, `cbnz`, `tbz`, `tbnz`, etc.)
- l'opération 5 supporte les caractères UTF-8 (par exemple: `aéケ` → `aéケ aケé éaケ éケa ケéa ケaé`)

Code partiel.

```
.global main

main:
    adr    x0, fmtStr
    adr    x1, chaîne
    bl     scanf

    mov    x0, 0
    bl     exit

.section ".data"
// Mémoire allouée pour une chaîne de caractères d'au plus 1024 octets
chaîne:   .skip 1024

.section ".rodata"
// Format pour lire une chaîne de caractères d'une ligne (incluant des espaces)
fmtStr:   .asciz "[%[^\\n]s"
```