

IFT209 – Programmation système
 Université de Sherbrooke
Laboratoire 4

Enseignant: Michael Blondin
 Date de remise: dimanche 24 février 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. Le but de ce laboratoire est d'implémenter l'algorithme de recherche dichotomique suivant:

Entrées : un tableau t d'entiers signés triés, la taille n du tableau, et un entier signé x
Sorties : un indice i tel que $t[i] = x$ s'il en existe un, sinon n

```

lo ← 0
hi ← n

tant que lo < hi faire
  mid ← (lo + hi) ÷ 2

  si t[mid] = x alors
    retourner mid                                /* Élément trouvé */
  sinon si t[mid] < x alors
    lo ← mid + 1                                /* Continuer recherche à droite */
  sinon
    hi ← mid                                    /* Continuer recherche à gauche */

retourner n                                    /* Élément non trouvé */

```

Cet algorithme permet de chercher un élément x dans un tableau trié t en temps logarithmique. Si le tableau contient x , alors un indice i tel que $t[i] = x$ est retourné; sinon la taille n du tableau est retournée.

Tests. Considérons les tableaux suivants dont les éléments sont des entiers signés de 32 bits:

```

tab0 = [2, 4, 6, 8, 10]
tab1 = [4, 205]
tab2 = [-100, -28, -15, 0, 7, 10, 15, 43, 76, 99, 100, 205, 1000]

```

Votre programme doit lire un entier signé x de 32 bits et afficher le résultat de la recherche de x sur chacun des tableaux ci-dessus. Par exemple, sur les trois tests suivants vous devriez obtenir:

4	205	-28
1	5	5
0	1	2
13	11	1

où, pour chaque test, la première ligne (en noir) est l'entrée, et les trois autres lignes (en cyan) sont la sortie.

Directives.

- Votre programme doit être obtenu en complétant le code partiel ci-bas;
- Votre programme doit être remis dans un seul fichier nommé `labo4.s`;
- Ne modifiez pas le point d'entrée ainsi que le contenu des tableaux;
- Vous pouvez supposer que les valeurs en entrée sont valides;
- Afin de vous préparer pour l'examen, tentez de ne pas copier/coller du code des autres laboratoires.

Pointage. Vous pouvez obtenir un maximum de 10 points. Vous obtenez:

- 1 point si votre programme lit un entier;
- 1 point si votre programme affiche trois entiers;
- 1 point si vous faites bien trois appels à `find` dans `main`;
- 3 points si votre programme passe les trois tests ci-dessus;
- 1 point si votre programme donne la bonne sortie sur d'autres entrées et d'autres tableaux choisis à la correction;
- 1,5 points si votre code est bien indenté (codes d'opération, opérandes et commentaires alignés);
- 1,5 points pour la présence de commentaires significatifs facilitant la lecture et compréhension du code.

Code partiel.

```
.global main

.macro SAVE
    stp    x29, x30, [sp, -96]!
    mov    x29, sp
    stp    x27, x28, [sp, 16]
    stp    x25, x26, [sp, 32]
    stp    x23, x24, [sp, 48]
    stp    x21, x22, [sp, 64]
    stp    x19, x20, [sp, 80]
.endm

.macro RESTORE
    ldp    x27, x28, [sp, 16]
    ldp    x25, x26, [sp, 32]
    ldp    x23, x24, [sp, 48]
    ldp    x21, x22, [sp, 64]
    ldp    x19, x20, [sp, 80]
    ldp    x29, x30, [sp], 96
.endm

main:
    // Lire un entier signé x de 32 bits

    // Afficher le résultat de find(&tab0, 5, x)
    // Afficher le résultat de find(&tab1, 2, x)
    // Afficher le résultat de find(&tab2, 13, x)

    bl    exit
```

```
find:
    /*
       code ici
    */

/*
   données ici
  */

.section ".data"
tab0: // tableau tab0 ici
tab1: // tableau tab1 ici
tab2: // tableau tab2 ici
```