

ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΦΟΙΤΗΤΗ:.....

ΑΡ. ΜΗΤΡΩΟΥ:

ΒΑΘΜΟΣ:

ΕΡΓΑΣΤΗΡΙΟ 03- 11/03/2020

Δημιουργήστε έναν φάκελο με όνομα **erg03_student-name_tp**, όπου **student** βάζετε το ονοματεπώνυμό σας και όπου **tp** τον αριθμό μητρώου σας στη σχολή.

1. Άσκηση 1

- Αντιγράψτε τον παρακάτω κώδικα σε ένα text editor. Ο κώδικας διαβάζει 2 αριθμούς a και b σαν είσοδο από το χρήστη και εκτυπώνει το άθροισμα, τη διαφορά και το γινόμενό τους:

```
.data    # initialization of data memory with the strings needed:
str_a:   .asciiz "a = "
str_b:   .asciiz "b = "
str_add: .asciiz "a + b = "
str_sub: .asciiz "a - b = "
str_mult: .asciiz "a * b = "
str_nl:  .asciiz "\n"      #Character for new line insertion

.text
.globl main
main:

    # PRINT prompt for a
    addi $2, $0, 4          # system call code for print_string
    la   $4, str_a          # pseudo-instruction: address of string
    syscall                # print the string from str_a
    # READ a
    addi $2, $0, 5          # system call code for read_int
    syscall                # read a line containing an integer
    add  $16, $2, $0        # copy returned int from $2 to a
    # PRINT prompt for b
    addi $2, $0, 4          # system call code for print_string
    la   $4, str_b          # pseudo-instruction: address of string
    syscall                # print the string from str_b
    # READ b
    addi $2, $0, 5          # system call code for read_int
    syscall                # read a line containing an integer
    add  $17, $2, $0        # copy returned int from $2 to b

    # Calc a+b
    add $18, $16, $17
    # PRINT prompt for a+b
    addi $2, $0, 4          # system call code for print_string
    la   $4, str_add        # pseudo-instruction: address of string
    syscall                # print the string from str_add
    # PRINT a+b
    addi $2, $0, 4          # system call code for print_string
    la   $4, str_mult        # pseudo-instruction: address of string
    syscall                # print the string from str_mult
    add  $18, $16, $17
    addi $2, $0, 4          # system call code for print_string
    la   $4, str_add        # pseudo-instruction: address of string
    syscall                # print the string from str_add
    add  $18, $16, $17
    addi $2, $0, 4          # system call code for print_string
    la   $4, str_mult        # pseudo-instruction: address of string
    syscall                # print the string from str_mult
```

```
    addi $2, $0, 1    # system call code for print_int
    add $4,$0,$18
    syscall
    # PRINT new line
    addi $2, $0, 4
    la $4, str_nl
    syscall

    # Calc a-b
    sub $19, $16, $17
    # PRINT prompt for a-b
    addi $2, $0, 4
    la $4, str_sub
    syscall
    # PRINT a-b
    addi $2, $0, 1    # system call code for print_int
    add $4,$0,$19
    syscall
    # PRINT new line
    addi $2, $0, 4
    la $4, str_nl
    syscall

    # Calc a*b
    mult $16, $17    #multiplies $16*$17 and stores the result in $LO
    mflo $20         #moves $LO to $20
    # PRINT prompt for a*b
    addi $2, $0, 4
    la $4, str_mult
    syscall
    # PRINT a*b
    addi $2, $0, 1    # system call code for print_int
    add $4,$0,$20
    syscall
    # PRINT new line
    addi $2, $0, 4
    la $4, str_nl
    syscall
j main
```

1.1 Αποθηκεύστε τον κώδικά με όνομα **erg03_ask01.s** μέσα στο φάκελο που δημιουργήσατε στην αρχή και στη συνέχεια φορτώστε το αρχείο μέσα στο πρόγραμμα **Pcspim** και εκτελέστε το. Στις τιμές των μεταβλητών, δώστε **a=7** και **b=3**.

1.2 Μετά την ολοκλήρωση της εκτέλεσης του προγράμματος και με τα αποτελέσματα να είναι όλα εμφανή στην κονσόλα, κάντε **print screen** το στιγμιότυπο και αποθηκεύστε το στο φάκελο που δημιουργήσατε στην αρχή με όνομα **capture01.jpg**.

2. Άσκηση 2

2.1 Ακολουθώντας την παρακάτω οδηγία, να προσθέσετε στο προηγούμενο πρόγραμμα τις απαραίτητες γραμμές κώδικα, ώστε εκτός του αθροίσματος, της διαφοράς και του γινομένου, να εμφανίζει και το ακέραιο **πηλίκο (div)** και **υπόλοιπο (mod)** των **a** και **b**. Συγκεκριμένα:

- η τιμή του $a \div b$ να αποθηκεύεται στον καταχωρητή \$21
- η τιμή του $a \bmod b$ να αποθηκεύεται στον καταχωρητή \$22

Αποθηκεύστε τον κώδικά με νέο όνομα (**erg03_ask02.s**) μέσα στο φάκελο που δημιουργήσατε στην αρχή και εκτελέστε τον στο PCsim για **a=8** και **b=5**. Θα πρέπει να εμφανιστούν τα ακόλουθα:

```
a = 8
b = 5
a + b = 13
a - b = 3
a * b = 40
a div b = 1
a mod b = 3
```

ΟΔΗΓΙΑ: Για να βρούμε το ακέραιο πηλίκο και υπόλοιπο δύο αριθμών **a** και **b**, χρησιμοποιούμε την εντολή **div κ1,κ2**, όπου κ1 και κ2 είναι οι δύο καταχωρητές στους οποίους είναι αποθηκευμένες οι τιμές των **a** και **b**. Η εκτέλεση της εντολής έχει σαν αποτέλεσμα να αποθηκεύεται στον καταχωρητή \$LO το πηλίκο του a/b και στον καταχωρητή \$HI το υπόλοιπο του a/b . Οι εντολές **mflo** και **mfhi** μας βοηθάνε να περάσουμε το περιεχόμενο των καταχωρητών \$LO και \$HI αντίστοιχα σε άλλους καταχωρητές, ώστε να είναι δυνατή η περαιτέρω πρόσβαση και επεξεργασία τους, π.χ. η εντολή **mflo \$25** θα περάσει το περιεχόμενο του \$LO στον καταχωρητή \$25.

2.2 Τα αποτελέσματα να αποθηκευτούν με νέο print screen στο φάκελο που δημιουργήσατε εξ' αρχής με όνομα **capture2.jpg**.

- ❖ Συμπίεστε σε μορφή **zip** το φάκελό σας και ανεβάστε το συμπιεσμένο αρχείο στο **eclass**, στην κατηγορία **Εργασίες ➔ Εργασία 03 (Τε_11-03-2020)**.

Σημείωση: Αν μέσα στο συμπιεσμένο αρχείο, ο φάκελος με τις ασκήσεις δεν έχει την κατάλληλη ονομασία (όπως υποδεικνύεται στην αρχή), δεν θα είναι δυνατή η βαθμολόγησή σας!

Καλή επιτυχία!