

Lab 3

Submit your work to moodle before the deadline

1. Implement a **‘replace’** procedure (function) in MIPS assembly language that, given an array of integers *Arr*, its *length*, integers *x* and *y*, replaces all *x* with *y* in *Arr*. Then your program should print out all values of *Arr*.

For example, if *Arr* = {21, 20, 51, 83, 20, 20}, *length* = 6, *x* = 20, *y* = 5 and *index* = 0, then after running your program the values of *Arr* MUST be *Arr* = {21, 5, 51, 83, 5, 5} and the values MUST be printed out. **Your procedure must follow the MIPS procedure call conventions.**

In the program, we assume the variables (e.g, *Arr*, *length*, *x*, *y* and *index*) should be declared and initialized manually in the **.data** section.

The signature of this procedure in a high level language would look like this:

```
void replace(int Arr[], int length, int x, int y);
```

2. Additionally, implement a **‘printArrInt’** procedure in MIPS assembly language that prints each element of the *Arr* using its *length*. Call this function twice to print the *Arr*. (i.e., before and after the **replace** procedure call).

The signature of this procedure in a high level language would look like this:

```
void printArrInt(int Arr[], int length);
```

Output:

21 20 51 83 20 20

21 5 51 83 5 5

NOTES: How to print Integers and Strings/space/newline using ‘syscall’

```
.data
x:          .word    5
msg1:       .asciiz  "x="
nl:         .asciiz  "\n"
space:      .asciiz  " "
```

```
.text
main:
    # Register assignments
    # $s0 = x

    # Initialize registers
    lw      $s0, x          # Reg $s0 = x

    # Print msg1
    li      $v0, 4          # print_string syscall code = 4
    la      $a0, msg1
```

syscall

Print result (x)

```
li      $v0,1          # print_int syscall code = 1
move    $a0, $s0       # Load integer to print in $a0
syscall
```

Print newline

```
li      $v0,4          # print_string syscall code = 4
la      $a0, nl
syscall
```

Exit

```
li      $v0,10         # exit
syscall
```