IST-ASM Final Exam — Fall 2022

Name:

- First, write your name in the box above. Then, have a quick read through all 5 questions.
- In the end, you will write up your answers on this paper.
  - But please make a draft elsewhere first. Only hand in something readable.
- This is an open-book open-laptop exam: you may work on scrap paper or on your screen.
- Each questions is independent from others, except stated otherwise.

Question 1 Perform the binary addition 77 + 43: convert both numbers to binary, then compte the sum entirely in binary. Show the details of your work.

```
        1  1  1  1

    1  0  0  1  1  0  1
 +     1  0  1  0  1  1
-----------------
    1  1  1  1  0  0  0  = 120
```

Question 2 Convert the program below to ASM syntax.

machine code (hex) ⟷ source program (asm)

```
00  209003e8
04  21990001
08  3209fffc
```

```
addi  r9, zero, 1000
subi  r9, r9, 1
blt   zero, r9, -4
```

```
    leti r9, 1000
loop:
    dec r9
    bgtz r9, loop
```

Question 3 Write a program which raises a number N to a power P. The idea is to multiply N by itself P times: $N \times N \times ... \times N$. Initially N and P are stored in R1 and R2, respectively. Both are assumed to be strictly positive.

```
        leti R1, 5
        leti R2, 4
        ;; 5**4 = 625

        leti R3, 1 ;; result
loop:
        beqz R2, done
        mul  R3, R3, R1
        dec  R2
        bra  loop

done:    bra done
```

Question 4 Write a program which fills the left half of the screen in yellow.

```
        leti R11, 0xB0000000 ;; base address of BRAM buffset
        leti R12, 0xFFFF0000 ;; RGB triplet for yellow

        leti R1, 0 ;; Y offset
y_loop:
        leti R2, 0 ;; X offset

        add R10, R11, R1 ;; base address of our line of pixels
x_loop:
        add R9, R10,  R2 ;; address of pixel
        store [R9], R12

        addi R2, R2, 4
        leti R3, 160   ;; 80/2 = 40 pixels, 4 bytes each
        blt  R2, R3, x_loop

        addi R1, R1, 320
        leti R3, 19200 ;; 60 lines, 80*4 = 320 bytes each
        blt R1, R3, y_loop
done:
        bra +0
```

Question 5 Definition: the *decimal digital root* of a natural number is defined as the value obtained by repeatedly summing the decimal digits of *N* until a single-digit number is reached. For instance, the decimal digital root of number 12345 is 6 because 1+2+3+4+5 = 15 and 1+5 = 6.

Write a recursive `ddr` function which computes the decimal digital root of a positive integer *N*:

- if $N<10$ then `ddr`(N) = N
- if $N\geqslant 10$ then `ddr`(N) = `ddr`$\big((N \div 10) + (N \bmod 10)\big)$
  for instance `ddr`(12345) = `ddr`(1234 + 5) = `ddr`(1239) = `ddr`(123 + 9) = ...

Notes: You'll want to use `DIV`/`DIVI` and `MOD`/`MODI` instructions to get the quotient and remainder of the integer division, respectively.

```
    leti SP, 0x10000000
main:
    leti R1, 12345
    call ddr
    bra +0

ddr:
```

```
        leti SP, 0x10000000
main:
        leti R1, 12345
        call ddroot
        bra +0

ddroot:
        push LR
        push R2
        push R3
        push R4

        leti R2, 10
        blt R1, R2, done

        div R3, R1, R2 ; R3 = N/10
        mod R4, R1, R2 ; R4 = N%10

        add R1, R3, R4
        call ddroot

done:
        pop R4
        pop R3
        pop R2
        pop LR
        ret
```