

Contrôle continu 1 (préparation) - Durée 15 min

Quelques exercices corrigés pour réviser

1 Un exercice d'assembleur

Écrire un code assembleur SARUMAN qui calcule la taille d'une chaîne de caractère stockée en mémoire.

On vous fournit le code à remplir, avec des commentaires.

Listing 1 – minmax.s

```
1      ;;
      lea r0 str          ; l'adresse du label str
strlen:
      leti r1 0          ;
      setctr a0 r0       ; a0 contient l'adresse du premier char
6 loop:
      ;; TODO (5 lignes)

11
end:
      print string "and the result is: "
      print signed r1
      ;; mettre le résultat en mémoire au label res
16      ;; 3 lignes – considérer que le résultat tient sur 16bits

21 fin:
      jump fin          ;boucle infinie pour le simul
      jump fin
str:
      .string "Hello, world!"
res:
26      .const 16 #100
```

Solution:

Listing 2 – minmax.s

```
      lea r0 str          ; the address of label str
strlen:
      ;
```

```

4      leti r1 0
      setctr a0 r0          ; r0 has the parameter @str
loop:
      readse a0 8 r2       ; read 8 bits in memory (a char)
      add r2 r2 0
      jumpif z end        ; if the char is 0, end.
9      add r1 1
      jump loop
end:
      print signed r1
      ;; now store in memory
14     lea r3 res
      setctr a1 r3
      write a1 16 r1

fin:
19     jump fin
str:
      .string "Hello, world!"
res:
      .const 16 #100

```

2 Deux corrections de TP1

2.1 Min et max en mémoire

Solution:

Listing 3 – minmax.s

```

leti r1 1232342342342342 ; first integer
2  leti r2 4523423423 ; second integer

lea r0 min                ; adress of the begining of the stack
setctr a0 r0              ; from this we can safely store values

7  cmp r1 r2
jumpif sgt swap ; If the integers are not in the right order, we swap then

write a0 64 r1 ; We write the min at mem[a0]
write a0 64 r2 ; We write the max at mem[a0] (a0 has been incremented by 64)
12 jump end

swap:
17  write a0 64 r2
      write a0 64 r1

```

```

end:
    setctr a0 r0
    readze a0 64 r3 ; We read the min from memory
22    readze a0 64 r4 ; We read the max from memory
    print signed r3 ; We print the min
    print char '\n'
    print signed r4 ; We print the max
    print char '\n'
27
min:

```

2.2 Dessiner les carrés d'étoile

Solution:

Pour les carrés, il s'agit de traduire le pseudo-code : (on suppose que N est dans le registre R1)

```

R2 <- -R1
R3 <- 0
while( R3 < R1 ) {
    R4 <- 0
    while( R4 < R1 ) {
        print('*'); // and shift 15 pts right
        R4++
    }
    print('\n');
    R3++;
}

```

Listing 4 – carres.s

```

leti r0 0 ; zero
2 leti r1 4 ; n
leti r2 0
sub2 r2 r1
leti r3 0

7 loop: ; boucle externe sur r3
    cmp r3 r1
    jumpif ge halt
    leti r4 0

12 star: ; boucle interne qui dessine les *
    cmp r4 r1 ; compteur = r4
    jumpif ge endline
    print char '*' ; et hop, dessine l'étoile
    add2i r4 1 ; incrémente r4
17 jump star

endline:

```

22

```
print char '\n'           ; à la fin de la ligne, on imprime...  
add2i r3 1              ; un saut de ligne, puis on incrémente r3  
jump loop
```

halt:

```
jump -13
```
