Démonstration 3

IFT6800

25 septembre 2015

1 Programme mystère

Le programme suivant, lorsqu'exécuté passe par les lignes `10,11,01,02,03,30` et termine.

```
program pleindesauts
// Input:    Rien
// Output:   Rien
// Remarks:  Le programme saute à la ligne 01, puis 30
// ------------------------------------------------------------------

01: 1000    no-op
02: 1000    no-op
03: C030    goto 30

10: 1000    no-op
11: C001    goto 01

30: 0000    halt
```

2 Compte à rebours "brute force"

Défi : Bâtir un programme qui imprime `0005`, `0004`, `0003`, `0002`, `0001`, `0000` à l'écran et arrête. Vous n'avez pas à faire de boucle, faites la tâche le plus simplement possible.

Exemple de solution :

```
program CompteARebours
// Input:    Rien
```

```
// Output:   0005, 0004, 0003, 0002, 0001, 0000
// ------------------------------------------------------------------

10: 7105    R[1] <- 0005
11: 91FF    mem[FF] <- R[1]
12: 7104    R[1] <- 0004
13: 91FF    mem[FF] <- R[1]
14: 7103    R[1] <- 0003
15: 91FF    mem[FF] <- R[1]
16: 7102    R[1] <- 0002
17: 91FF    mem[FF] <- R[1]
18: 7101    R[1] <- 0001
19: 91FF    mem[FF] <- R[1]
1A: 7100    R[1] <- 0000
1B: 91FF    mem[FF] <- R[1]
1C: 0000    halt
```

3 Compte à rebours avec boucle

Défi : Bâtir un programme qui imprime 0005, 0004, 0003, 0002, 0001, 0000 à l'écran et arrête. Utilisez une boucle de manière à ce que vous n'ayez qu'une seule instruction qui accède au stdout.

Exemple de solution :

```
program CompteARebours Boucle
// Input:    Rien
// Output:   0005, 0004, 0003, 0002, 0001, 0000
// ------------------------------------------------------------------

10: 7A05    R[A] <- 0005
11: 7101    R[1] <- 0001
12: 9AFF    write(R[A])
13: DA15    if (R[A] > 0) goto 15
14: 0000    halt
15: 2AA1    R[A] <- R[A] - R[1]
16: C012    goto 12
```

4 Compte à rebours avec saut

Défi : Bâtir un programme qui prend en entrée deux entiers strictements positifs a et b et qui affiche a, $a - b$, $a - 2b$, ... jusqu'à ce qu'on arrive à un nombre plus petit ou égal à 0.

Exemple de solution :

```
program CompteARebours Input
// Input:    a, b des entiers positifs
// Output:   le compte à rebours à partir de a avec sauts de b jusqu'à
//           ce qu'on se rende à un nombre <= 0
// ------------------------------------------------------------------

10: 8AFF    read R[A]
11: 8BFF    read R[B]
12: 9AFF    write(R[A])
13: DA15    if (R[A] > 0) goto 15
14: 0000    halt
15: 2AAB    R[A] <- R[A] - R[B]
16: C012    goto 12
```

5 Plus grand ou égal

Défi : Bâtir un programme qui détermine si le premier entier entré par l'usager est plus grand ou égal au deuxième. Le programme doit mettre 1 dans stdout si c'est le cas et 0 sinon.

Exemple de solution :

```
program PlusGrandEgal
// Input:    Deux entiers a et b en hexadécimal
// Output:   1 si a >= b et 0 sinon
// ------------------------------------------------------------------

10: 8AFF    read R[A]
11: 8BFF    read R[B]
12: 7101    R[1] <- 0001
13: 2CAB    R[C] <- R[A] - R[B]
14: 1CC1    R[C] <- R[C] + R[1]
```

```
15: DC18    if (R[C] > 0) goto 18
16: 90FF    write R[0]
17: 0000    halt
18: 91FF    write R[1]
19: 0000    halt
```