



DIRO
IFT 1215

DÉMONSTRATION N° 6

– Correction –

Max Mignotte

DIRO, Département d'Informatique et de Recherche Opérationnelle, local 2377.

[http: //www.iro.umontreal.ca/~mignotte/ift1215/](http://www.iro.umontreal.ca/~mignotte/ift1215/)

E-mail: mignotte@iro.umontreal.ca

LMC

Exercise 6.7

(BL3-) This problem forces students to consider a more general approach. The simplest approach is to create a location for the highest number and load the first number into it. Each following number is read in, stored (in case it's needed), and compared. The higher number is stored, and the process continued until the 00 flag is encountered.

00	INP	901	
01	STO 99	399	;store the first number in 99
02	INP	901	
03	BRZ 11	711	;completion flag
04	STO 98	398	;store new value temporarily
05	SUB 99	299	
06	BRP 08	808	;new value larger--replace
07	BR 02	602	;old value larger--get next number
08	LDA 98	598	
09	STO 99	399	
10	BR 02	602	
11	LDA 99	599	
12	OUT	902	
13	COB	000	

Exercise 6.8

(BL3) This problem is best offered as a challenge to students with strong programming skills. It requires careful analysis, and takes lots of steps regardless of the approach. Here's one solution. It puts the first two values in order first, then compares the third value against each.

```

00 INP      901
01 STO 99   399
02 INP      901
03 STO 98   398
04 SUB 99   299
05 BRP 12   812      ;numbers are in order (98, 99)
06 LDA 99   599      ;swap 98 and 99
07 STO 96   396      ;a common error is to forget to save the...
08 LDA 98   598      ;...first number in the swap
09 STO 99   399
10 LDA 96   596
11 STO 98   398
12 INP      901      ;input third number
13 STO 97   397
14 SUB 98   298
15 BRP 30   830      ;numbers are in order (97, 98, 99)
16 LDA 97   597      ;swap 97 and 98
17 STO 96   396      ;note: a trick--swapping this way leaves 98 in A...
18 LDA 98   598      ;...ready for next operation
19 STO 97   397
20 LDA 96   596
21 STO 98   398
22 SUB 99   296      ;compare 98 and 99
23 BRP 30   830      ;(98, 99) are in correct order
24 LDA 98   598      ;swap 98 and 99
25 STO 96   396
26 LDA 99   599
27 STO 98   398
28 LDA 96   596
29 STO 99   399
30 LDA 97   597      ;output results
31 OUT      902
32 LDA 98   598
33 OUT      902
34 LDA 99   599
35 OUT      902
36 COB      000

```

Most students will come up with solutions ranging from 40 to 70 instructions. I sometimes offer a prize (coffee and donuts) to the student coming up with the shortest solution.

Exercise 6.9

(BL2+) The trick here is in counting the number of times through the loop correctly. With only one accumulator, the sum must be saved each time before the index can be

manipulated. The student will need to figure out that the values “0” and “1” must be placed somewhere as constants to be used. The sum must be initialized to “0”, or the value will be incorrect if the program is rerun!

```

00 LDA 90    590
01 STO 99    399      ;initialize sum to zero
02 INP      901      ;input the count..
03 STO 98    399      ;..and save it in 98
04 INP      901      ;input a number
05 ADD 99    199
06 STO 99    399      ;add to sum and store sum
07 LDA 98    598
08 SUB 91    291      ;decrement count by 1
09 BRZ 11    711      ;done--output result
10 BR 04     704      ;not done--ready for next input
11 LDA 99    599
12 OUT      902      ;output result
13 COB
90 DAT 00    000      ;constant value 0
91 DAT 01    001      ;constant value 1

```

Exercise 6.16

(BL2+) As in 6.15, it is necessary to calculate a condition code that can use the branch instructions.

```

MM   program code to be repeated
     ....
     program code to calculate test condition result
     say, positive or 0 for TRUE, negative for FALSE
     ....
     BRP PP
     BR MM /FALSE, repeat the program code
PP   program code following REPEAT-UNTIL

```

Exercise 6.20

(BL2+) Suppose the branch is located at mailbox 23. Then the following sequence will do the job:

```

23 BRP 25
24 BR 50

```

Exercise 6.21

(BL2+) Suppose the branch is located at mailbox 23. Then the following sequence will work:

```

23 BRZ 25
24 BRP 75

```