UCF

School of Computer Science
CGS 3269 Computer Architecture
Summer 2005
DUE 7/6/05

Write an assembly program using Marie's instructions that implements the following pseudo codes:

array1[5] = {1, 3, 5, 7, 9};   *//store this two array in Marie's memory*
array2[5] = {2, 4, 6, 8, 10};

for i = 0 to 5
{
    array3[i] = array1[i] * array2[i];   *//multiply the corresponding elements in two*
                                         //arrays and store the result into the third
                                         //array
}   *// output every value in array3*

Output = 0;
for i = 0 to 5
{
    Output = array3[i] + Output;   *//summation of all the values in the third*
                                         //array3
}   *//output the summation*

The first array starts at address $(50)_{hex}$ in the memory, the second at $(60)_{hex}$, and the third at $(70)_{hex}$. And the program is loaded in Marie staring at address 100(in hex).

And when run the program, it will output every value in the third array, and the summation of them, as it is shown in Figure 1.

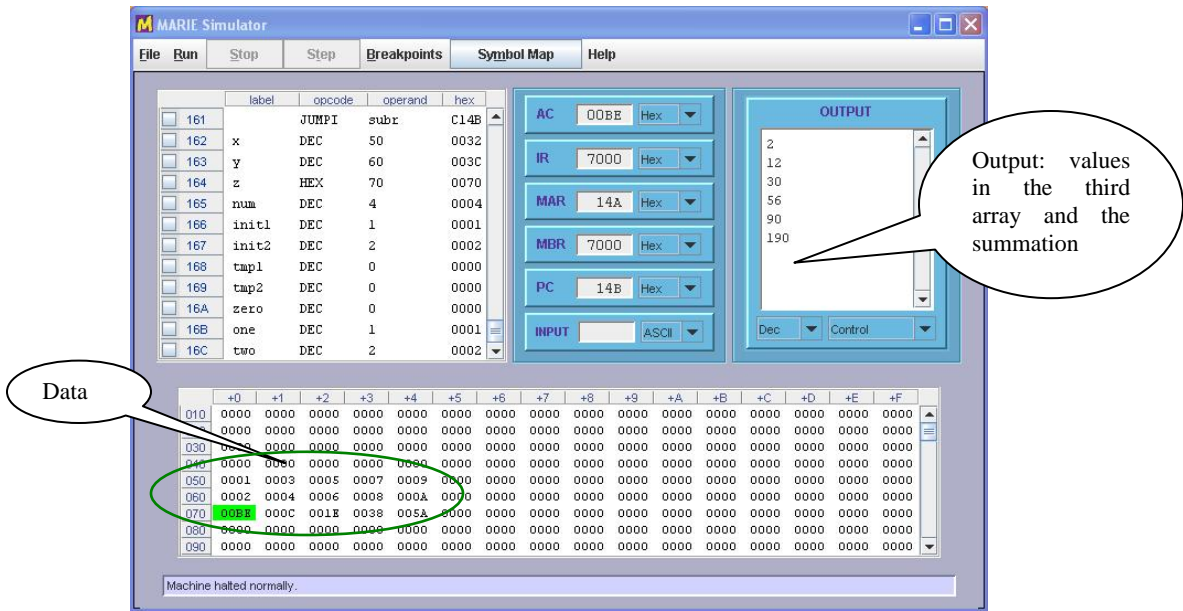Please turn in a floppy disk with your program in it, thanks.

MARIE Simulator

File Run Stop Step Breakpoints Symbol Map Help

	label	opcode	operand	hex
161		JUMPI	subr	C14B
162	x	DEC	50	0032
163	y	DEC	60	003C
164	z	HEX	70	0070
165	num	DEC	4	0004
166	init1	DEC	1	0001
167	init2	DEC	2	0002
168	tmp1	DEC	0	0000
169	tmp2	DEC	0	0000
16A	zero	DEC	0	0000
16B	one	DEC	1	0001
16C	two	DEC	2	0002

AC 00BE Hex
IR 7000 Hex
MAR 14A Hex
MBR 7000 Hex
PC 14B Hex
INPUT   ASCII

OUTPUT
2
12
30
56
90
190
Dec Control

Output: values in the third array and the summation

Data

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
010	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
020	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
030	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
040	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
050	0001	0003	0005	0007	0009	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
060	0002	0004	0006	0008	000A	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
070	00BE	000C	001E	0038	005A	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
080	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
090	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Machine halted normally.

Figure 1