

## Part 5 Instruction Statistics

### PART 1

#### Total part 1 statistics:

Total	1860
ALU	1065
Jump	8
Branch	268
Memory	515
Other	4

#### Set 16x16 (gen\_a):

Total	22
ALU	13
Jump	2
Branch	4
Memory	1
Other	2

#### Get 16x16 (TEST\_A\_16x16):

Total	21
ALU	12
Jump	2
Branch	4

Memory	1
Other	2

**Copy 16x16 (GEN\_B):**

Total	1801
ALU	1030
Jump	2
Branch	256
Memory	512
Other	1

**PART 2**

**Total part 2 Statistics:**

Total	1583
ALU	847
Jump	54
Branch	137
Memory	494
Other	51

**Sum neighbors (only one test 15 x 15):**

Total	515
ALU	276

Jump	18
Branch	42
Memory	163
Other	16

**Sum neighbors relate to get\_16x16 because it calls get\_16x16 in order to determine the number in memory. We call get\_16x16 8 times, skipping it once thanks to our flag (we don't want to add the middle value). This is why Sum neighbors' total value is over 8 times greater than get\_16x16.**

### **PART 3**

#### **Total part 3 Statistics:**

Total	24266
ALU	15232
Jump	1541
Branch	3872
Memory	2852
Other	769

#### **Bitmap to 16x16:**

Total	6010
ALU	3658
Jump	515

Branch	1296
Memory	284
Other	257

**Bitmap to 16x16 relates to set\_16x16. We can clearly see this in the numbers. Bitmap\_to\_16x16 calls set\_16x16 in order to set certain values to white, so our bitmap display can set what's given (turn black to white).**

**Draw 16x16:**

Total	18259
ALU	11575
Jump	1027
Branch	2576
Memory	2568
Other	513

**Draw 16x16 relates to both set16x16 and get\_16x16. This is evident from the fact that draw 16x16 has a lot more statistics than both of these procedures. This is where you physically draw the values. In order to do this you must get, then set the value of the address in the bitmap display. If the bit is 1, then it's set . If it's 0, it's unset.**