Jialiang Zhao
CSc 252

Part 1
Report of address of some variable.

La $t2, red 0x00400000
La $t3, orange 0x0040000c
La $t4, yellow 0x00400018
La $t5, green 0x00400024

Report of address of some variable.
Lui $1,0x00001001
ori $10, $1, 0x00000000   for red
Lui $1,0x00001001
ori $11, $1, 0x00000004   for orange
Lui $1,0x00001001
ori $12, $1, 0x00000008   for yellow
Lui $1,0x00001001
ori $10, $1, 0x0000000c   for green

Report of Hex encodings
0x3c011001
0x342a0000 for red

0x3c011001
0x342b0004 for orange

0x3c011001
0x342c0008 for yellow

0x3c011001
0x342d000c for green

Part2
I use EQUAL: .asciiz “EQUALS\n”

E:0x45
Q:0x51
U:0x55
A:0x41

0x10010060   for la $a0,EQUAL

0x41555145

I realized that the arrangement of addresses is arranged from the beginning of 0x000000. Every time you continue to add an element (using la), the address 0x00000 will get a new address. Through this method, the storage address of the processor can be visually seen. I learned that the original c and java we learned are the languages obtained after the ultimate simplification. The birth of each new language is a great product of the in-depth understanding of processors, addresses, and pointers by highly creative people. After using the code, when I already have an understanding of the address, I believe I can reduce the use of redundant addresses in writing the code. Used to increase the speed of the processor.