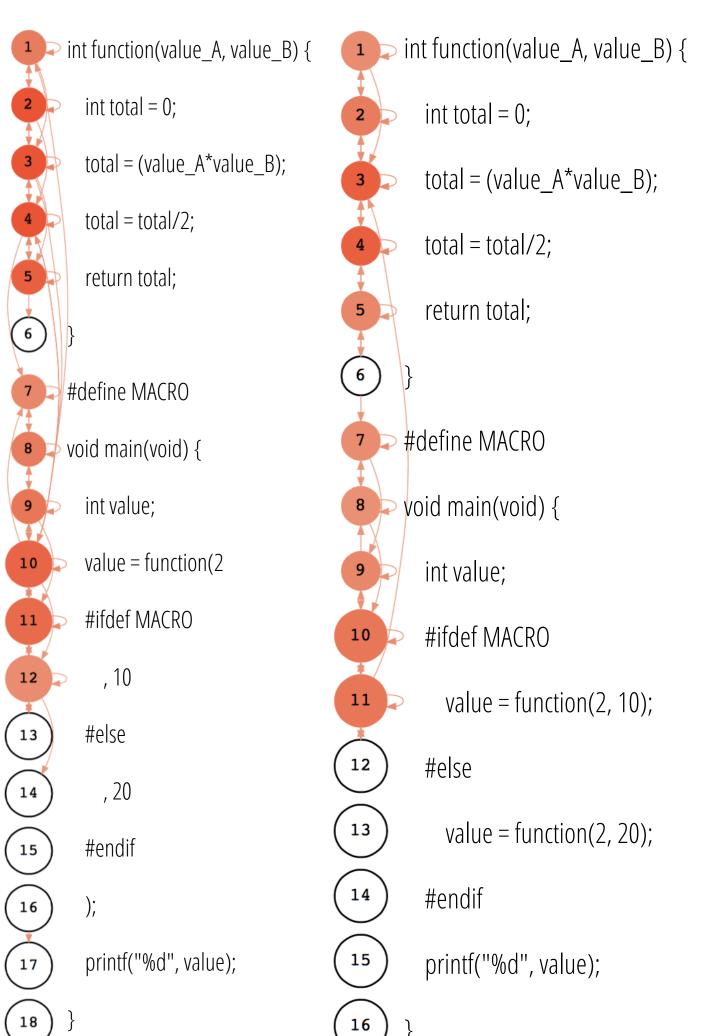
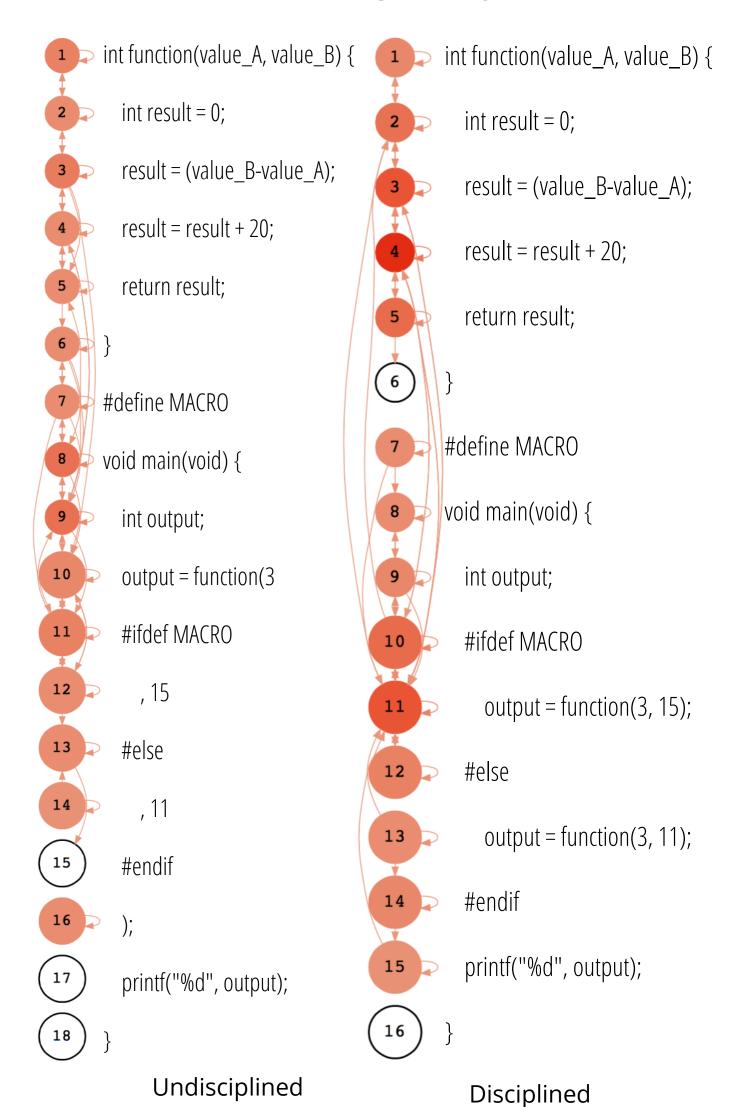
Refactoring 1 - Project 1



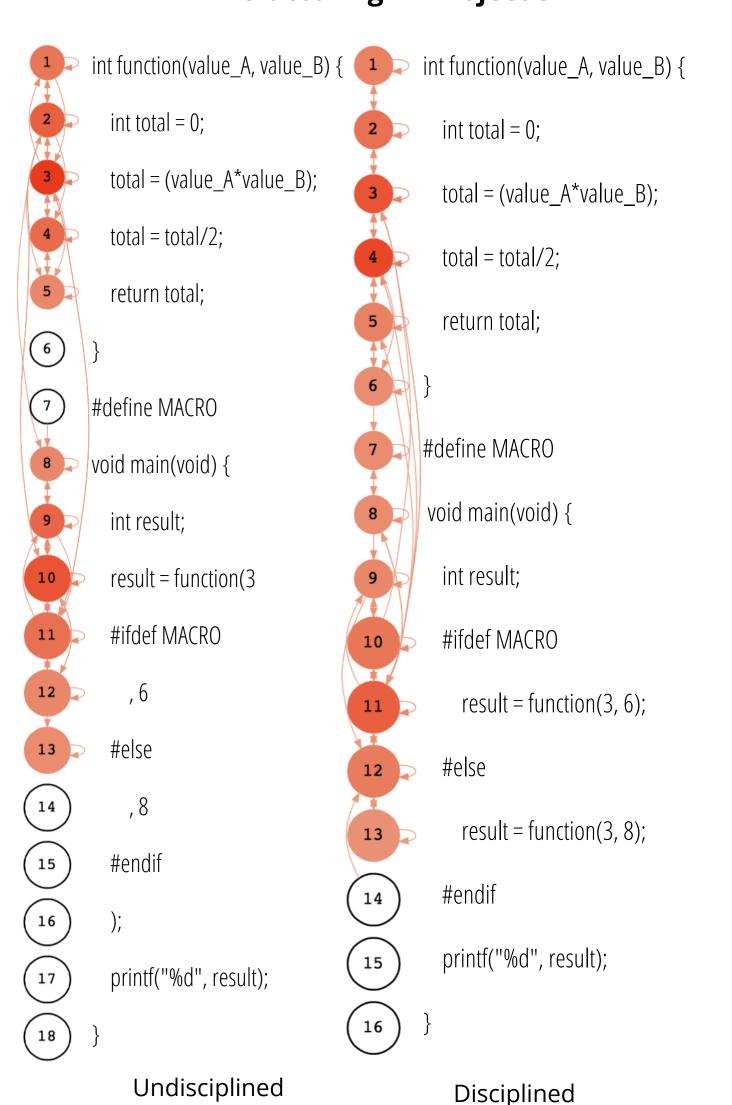
Refactoring 1 - Project 2



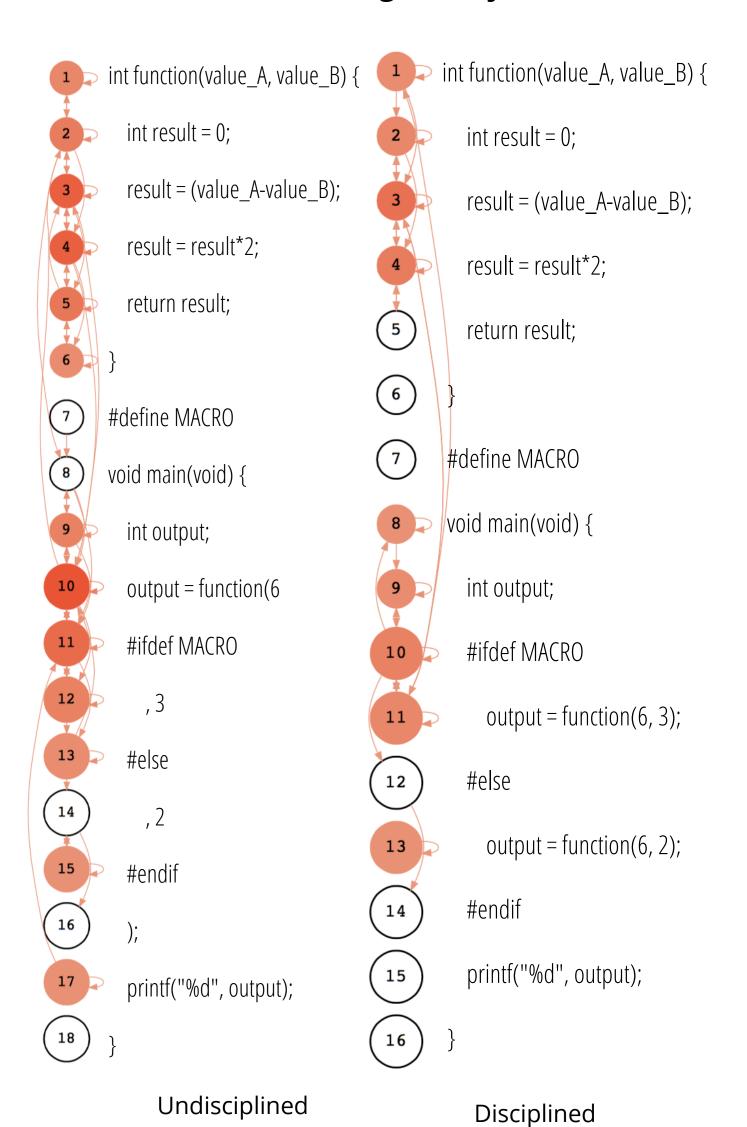
Refactoring 1 - Project 3

Undisciplined

Disciplined



Refactoring 1 - Project 4

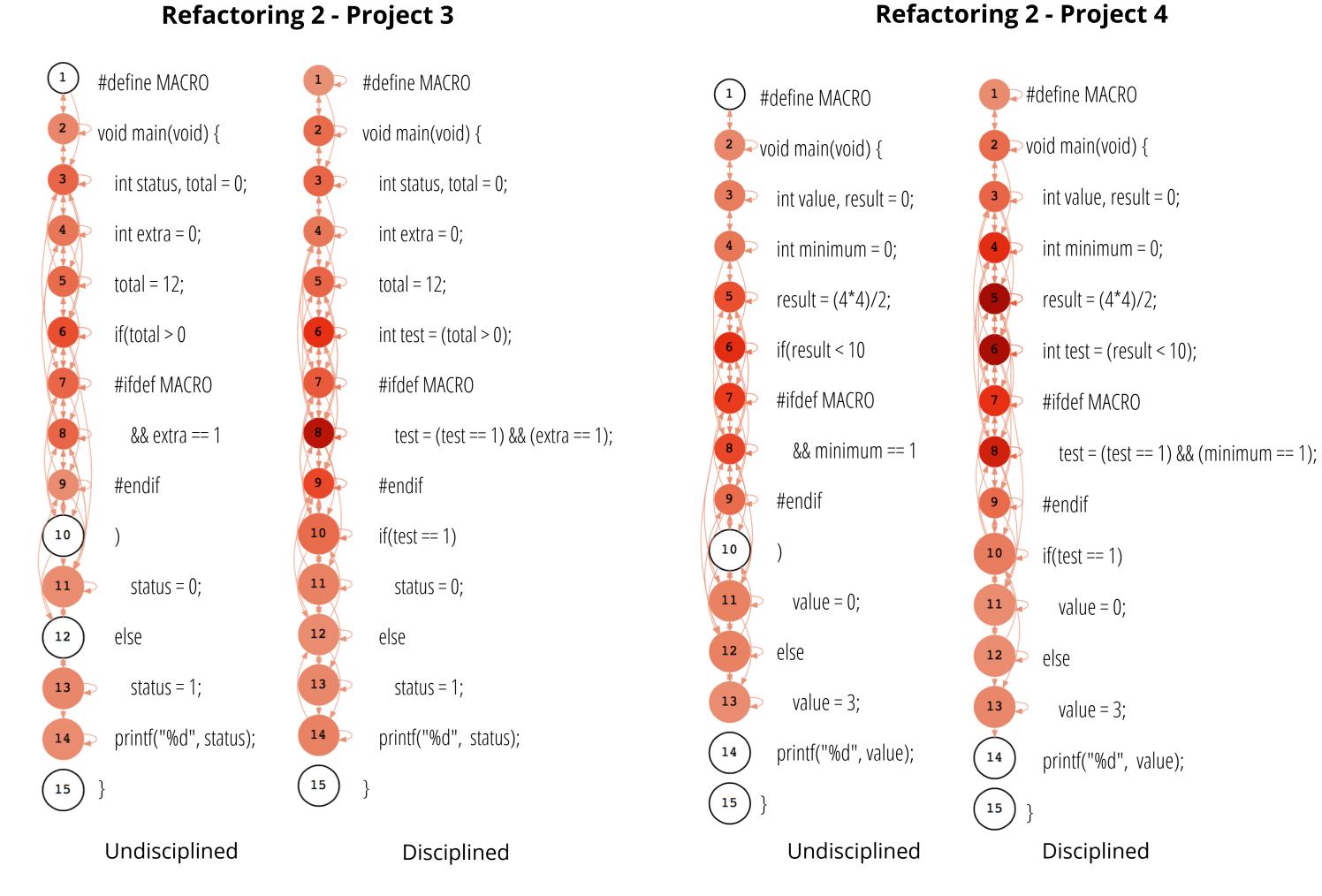


Refactoring 2 - Project 1

#define MACRO 1) #define MACRO #define MACRO #define MACRO void main(void) { void main(void) { void main(void) { void main(void) { int status, total = 0; int value, result = 0; int value, result = 0; int status, total = 0; int extra = 0; int minimum = 1; int minimum = 1; int extra = 0; total = 10; total = 10; result = (3 * 4)/2; result = (3 * 4)/2; if(total > 9 int test = (total > 9); if(result < 10 int test = (result < 10); #ifdef MACRO #ifdef MACRO #ifdef MACRO #ifdef MACRO && extra == 1 test = (test == 1) && (extra == 1); && minimum == 2 test = (test == 1) && (minimum == 2); #endif #endif #endif #endif if(test == 1)10 if(test == 1) status = 1; status = 1; value = 0; value = 0; else else else else status = 0; status = 0; value = 1; value = 1; printf("%d", status); printf("%d", status); printf("%d", value); 14 printf("%d", value); 15 (15) [15] Undisciplined Disciplined Disciplined Undisciplined

Refactoring 2 - Project 2

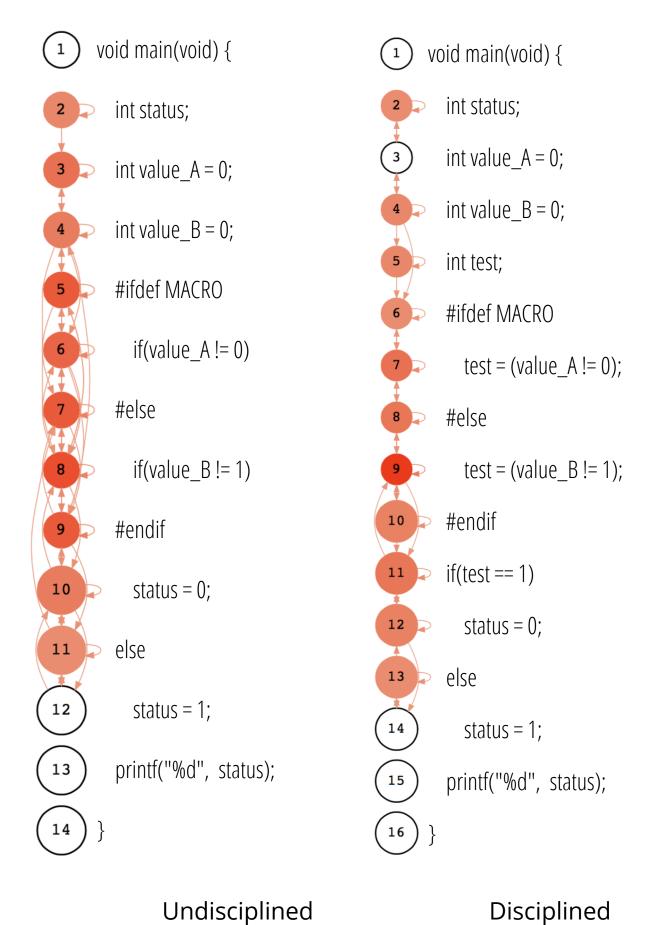
Refactoring 2 - Project 3



Refactoring 3 - Project 1

void main(void) { void main(void) { int output; int output; int value_A = 0; int value_A = 0; int value_B = 0; int value_B = 0; int test; #ifdef MACRO #ifdef MACRO if(value_A == 1) test = (value_A == 1); #else #else test = (value_B == 0); if(value_B == 0) #endif #endif if(test == 1) output = 1; 12 output = 1; else 13 else output = 0; 14 output = 0; 13 printf("%d", output); (15) printf("%d", output); 14) $\left(\begin{array}{c} 16 \end{array}\right)$

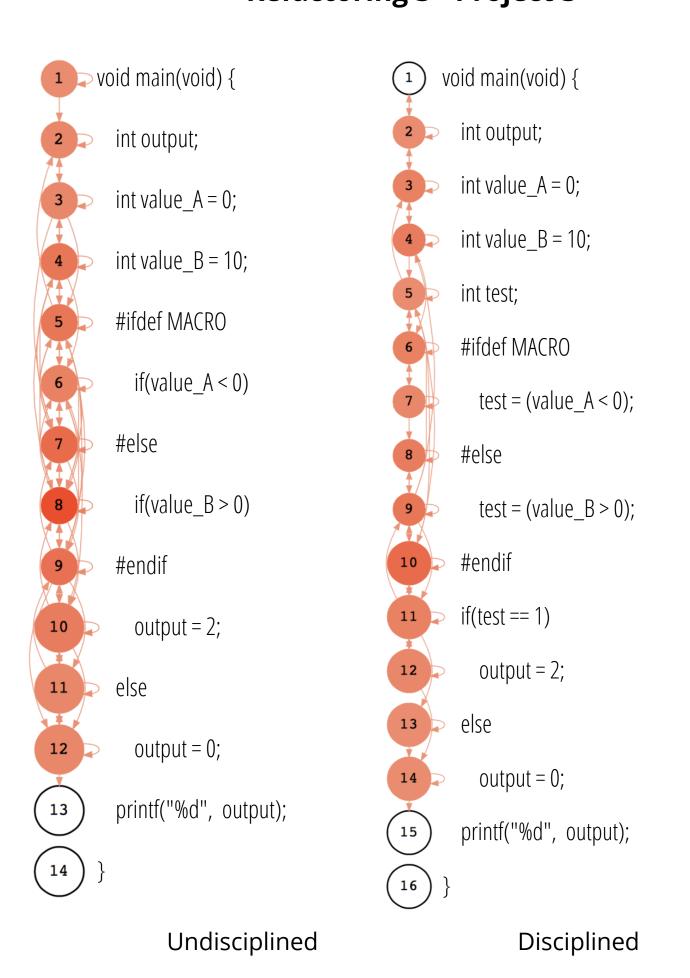
Refactoring 3 - Project 2



Refactoring 3 - Project 3

Disciplined

Undisciplined



Refactoring 3 - Project 4

