binary\_node\_t \*binary\_node = (binary\_node\_t \*) node;

if (binary\_node->op == '\*') {

                num\_node\_t \*num\_node\_right = (num\_node\_t \*) binary\_node->right;

                int64\_t value\_right = num\_node\_right->value;

                int i = 0;

                while (value\_right > 0) {

                    i++;

                    if (value\_right == 1) {

                        break;

                    }

                    if (value\_right %2 != 0) {

                        i = 0;

                        break;

                    }

                    value\_right = value\_right/2;

                }

                if (i != 0) {

                    compile\_ast(binary\_node->left);

                    printf("shl $%zu, %%rdi\n", i);

                    break;

                }

                else {

                    compile\_ast(binary\_node->left);

                    printf("pushq %%rdi\n");

                    compile\_ast(binary\_node->right);

                    printf("popq %%rax\n");

                    printf("imulq %%rax, %%rdi\n");

                    break;

                }

            }