

gcd.rus

Abstract Syntax Tree

```
1, 2
    0, 259 VAR
    0, 291 a
1, 2
    0, 259 VAR
    0, 291 b
1, 2
    0, 259 VAR
    0, 291 r
1, 10
    0, 273 =
    0, 291 a
    0, 292 64
1, 10
    0, 273 =
    0, 291 b
    0, 292 12
1, 6
    0, 268 DO
    0, 292 -1
    1, 1
        1, 10
            0, 273 =
            0, 291 r
            0, 291 a
        1, 10
            0, 273 =
            0, 291 a
            0, 291 b
        1, 10
            0, 273 =
            0, 291 b
            1, 10
                0, 278 %
                0, 291 r
                0, 291 b
            1, 3
                0, 260 IF
                1, 10
                    0, 284 <=
                    0, 291 b
                    0, 292 0
                1, 1
                    1, 4
                        0, 265 BREAK
```

1, 7

0, 269 PRINT

1, 8

0, 294 "gcd is "

0, 291 a

Generate LMC assembler Code

BOX _LOOP_TEMP

BOX _NUM_TEMP

BOX _LOGC_TEMP

BOX _CNT_TEMP

BOX _ITER_TEMP

BOX _REDIRECTION_FILE

BOX a

BOX b

BOX r

LDA #64

STA a

LDA #12

STA b

LDA #-1

STA _LOOP_TEMP

LDA #0

STA _CNT_TEMP

LOOP_LABEL_0:

LDA _CNT_TEMP

CMPA _LOOP_TEMP

JCOND BREAK_LABEL_0:

LDA a

STA r

LDA _CNT_TEMP

ADDA #1

JMP LOOP_LABEL_0

BREAK_LABEL_0:

prime.rus

Abstract Syntax Tree

```
1, 2
    0, 259 VAR
    0, 291 prime
1, 2
    0, 259 VAR
    0, 291 i
1, 2
    0, 259 VAR
    0, 291 flag
1, 9
    0, 290 OUTPUT
    0, 294 "prime_result.txt"
1, 10
    0, 273 =
    0, 291 prime
    0, 292 131
1, 10
    0, 273 =
    0, 291 flag
    0, 292 0
1, 3
    0, 260 IF
    1, 10
        0, 282 &&
        1, 10
            0, 284 <
            0, 292 2
            0, 291 prime
        1, 10
            0, 280 !=
            0, 292 0
            1, 10
                0, 278 %
                0, 291 prime
                0, 292 2
    1, 1
        1, 10
            0, 273 =
            0, 291 i
            0, 292 3
        1, 6
            0, 268 DO
            0, 292 -1
            1, 1
                1, 3
```

```

                                0, 260 IF
                                1, 10
                                    0, 284 <=
                                    0, 291 prime
                                    0, 291 i
                                1, 1
                                    1, 10
                                        0, 273 =
                                        0, 291 flag
                                        0, 292 1
                                    1, 4
                                        0, 265 BREAK
                                1, 3
                                    0, 260 IF
                                    1, 10
                                        0, 279 ==
                                        0, 292 0
                                        1, 10
                                            0, 278 %
                                            0, 291 prime
                                            0, 291 i
                                    1, 1
                                        1, 4
                                            0, 265 BREAK
                                1, 10
                                    0, 273 =
                                    0, 291 i
                                    1, 10
                                        0, 274 +
                                        0, 291 i
                                        0, 292 2
1, 3
    0, 260 IF
    1, 10
        0, 279 ==
        0, 292 1
        0, 291 flag
    1, 1
        1, 7
            0, 269 PRINT
            1, 8
                0, 291 prime
                0, 294 "is prime."
1, 3
    0, 260 IF
    1, 10
        0, 279 ==

```

```

        0, 292 0
        0, 291 flag
1, 1
        1, 7
            0, 269 PRINT
            1, 8
                0, 291 prime
                0, 294 "is not prime."

```

Generate LMC assembler Code

```

BOX    _LOOP_TEMP
BOX    _NUM_TEMP
BOX    _LOGC_TEMP
BOX    _CNT_TEMP
BOX    _ITER_TEMP
BOX    _REDIRECTION_FILE
BOX    prime
BOX    i
BOX    flag
LDA    #131
STA    prime
LDA    #0
STA    flag
STA    _LOGC_TMP
LDA    _LOGC_TMP
CMPA   #0
JCOND  IF_LABEL_0
LDA    #3
STA    i
IF_LABEL_0:
STA    _LOGC_TMP
LDA    _LOGC_TMP
CMPA   #0
JCOND  IF_LABEL_1
IF_LABEL_1:
STA    _LOGC_TMP
LDA    _LOGC_TMP
CMPA   #0
JCOND  IF_LABEL_2
IF_LABEL_2:

```

random.rus

Abstract Syntax Tree

```
1, 6
    0, 268 DO
    0, 292 10
    1, 1
        1, 7
            0, 269 PRINT
            1, 10
                0, 289 $
        1, 7
            0, 269 PRINT
            1, 10
                0, 288 RANDOM
                0, 292 1
                0, 292 10
```

Generate LMC assembler Code

```
BOX    _LOOP_TEMP
BOX    _NUM_TEMP
BOX    _LOGC_TEMP
BOX    _CNT_TEMP
BOX    _ITER_TEMP
BOX    _REDIRECTION_FILE
LDA    #10
STA    _LOOP_TEMP
LDA    #0
STA    _CNT_TEMP
LOOP_LABEL_0:
LDA    _CNT_TEMP
CMPA   _LOOP_TEMP
JCOND  BREAK_LABEL_0:
LDA    _CNT_TEMP
ADDA   #1
JMP    LOOP_LABEL_0
BREAK_LABEL_0:
```

sum.rus

Abstract Syntax Tree

```
1, 2
    0, 259 VAR
    0, 291 sum
1, 10
    0, 273 =
    0, 291 sum
    0, 292 0
1, 6
    0, 268 DO
    0, 292 10
    1, 1
        1, 10
            0, 273 =
            0, 291 sum
            1, 10
                0, 274 +
                0, 291 sum
                1, 10
                    0, 274 +
                    1, 10
                        0, 289 $
                        0, 292 1
            1, 7
                0, 269 PRINT
                0, 291 sum
```

Generate LMC assembler Code

```
BOX    _LOOP_TEMP
BOX    _NUM_TEMP
BOX    _LOGC_TEMP
BOX    _CNT_TEMP
BOX    _ITER_TEMP
BOX    _REDIRECTION_FILE
BOX    sum
LDA    #0
STA    sum
LDA    #10
STA    _LOOP_TEMP
LDA    #0
STA    _CNT_TEMP
LOOP_LABEL_0:
LDA    _CNT_TEMP
CMPA   _LOOP_TEMP
JCOND  BREAK_LABEL_0:
LDA    sum
```

```
STA    _NUM_TEMP
ADDA    #1
ADDA    _NUM_TEMP
STA    sum
LDA    _CNT_TEMP
ADDA    #1
JMP LOOP_LABEL_0
BREAK_LABEL_0:
```