1 1 1 CONST\_SYMBOL const

2 1 7 INT\_SYMBOL int

3 1 11 IDENT max\_value\_limit

4 1 27 BECOME =

5 1 29 INT 100

6 1 32 SEMICOLON ;

7 2 1 CONST\_SYMBOL const

8 2 7 INT\_SYMBOL int

9 2 11 IDENT min\_value\_limit

10 2 27 BECOME =

11 2 29 SUB -

12 2 30 INT 100

13 2 33 SEMICOLON ;

14 3 1 CONST\_SYMBOL const

15 3 7 INT\_SYMBOL int

16 3 11 IDENT max\_bounds\_limit

17 3 28 BECOME =

18 3 30 INT 10

19 3 32 SEMICOLON ;

20 4 1 CONST\_SYMBOL const

21 4 7 INT\_SYMBOL int

22 4 11 IDENT min\_bounds\_limit

23 4 28 BECOME =

24 4 30 INT 0

25 4 31 SEMICOLON ;

26 6 1 INT\_SYMBOL int

27 6 5 IDENT i

28 6 6 SEMICOLON ;

29 8 1 INT\_SYMBOL int

30 8 5 IDENT returnyear

31 8 15 LEFT\_BRACE {

32 9 5 RETURN\_SYMBOL return

33 9 12 LEFT\_PAR (

34 9 13 INT 2018

35 9 17 RIGHT\_PAR )

36 9 18 SEMICOLON ;

37 10 1 RIGHT\_BRACE }

38 12 1 INT\_SYMBOL int

39 12 5 IDENT add

40 12 8 LEFT\_PAR (

41 12 9 INT\_SYMBOL int

42 12 13 IDENT x

43 12 14 COMMA ,

44 12 16 INT\_SYMBOL int

45 12 20 IDENT y

46 12 21 RIGHT\_PAR )

47 12 22 LEFT\_BRACE {

48 13 5 RETURN\_SYMBOL return

49 13 12 LEFT\_PAR (

50 13 13 PLUS +

51 13 15 IDENT x

52 13 17 PLUS +

53 13 19 IDENT y

54 13 20 RIGHT\_PAR )

55 13 21 SEMICOLON ;

56 14 1 RIGHT\_BRACE }

57 16 1 INT\_SYMBOL int

58 16 5 IDENT gauss\_sum

59 16 14 LEFT\_PAR (

60 16 15 INT\_SYMBOL int

61 16 19 IDENT x

62 16 20 COMMA ,

63 16 22 INT\_SYMBOL int

64 16 26 IDENT y

65 16 27 RIGHT\_PAR )

66 16 28 LEFT\_BRACE {

67 17 5 CONST\_SYMBOL const

68 17 11 INT\_SYMBOL int

69 17 15 IDENT err\_max\_value\_limit

70 17 35 BECOME =

71 17 37 INT 1

72 17 38 SEMICOLON ;

73 18 5 CONST\_SYMBOL const

74 18 11 INT\_SYMBOL int

75 18 15 IDENT err\_min\_value\_limit

76 18 35 BECOME =

77 18 37 INT 2

78 18 38 SEMICOLON ;

79 19 5 CONST\_SYMBOL const

80 19 11 INT\_SYMBOL int

81 19 15 IDENT err\_invalid\_input

82 19 33 BECOME =

83 19 35 INT 5

84 19 36 SEMICOLON ;

85 21 5 IF\_SYMBOL if

86 21 7 LEFT\_PAR (

87 21 8 IDENT x

88 21 10 GTR >

89 21 12 IDENT y

90 21 13 RIGHT\_PAR )

91 22 9 IDENT error

92 22 14 LEFT\_PAR (

93 22 15 IDENT err\_invalid\_input

94 22 32 RIGHT\_PAR )

95 22 33 SEMICOLON ;

96 23 5 ELSE\_SYMBOL else

97 24 9 IF\_SYMBOL if

98 24 11 LEFT\_PAR (

99 24 12 IDENT y

100 24 14 GTR >

101 24 16 IDENT max\_value\_limit

102 24 31 RIGHT\_PAR )

103 25 13 IDENT error

104 25 18 LEFT\_PAR (

105 25 19 IDENT err\_max\_value\_limit

106 25 38 RIGHT\_PAR )

107 25 39 SEMICOLON ;

108 26 9 ELSE\_SYMBOL else

109 27 13 IF\_SYMBOL if

110 27 15 LEFT\_PAR (

111 27 16 IDENT x

112 27 18 LSS <

113 27 20 IDENT min\_value\_limit

114 27 35 RIGHT\_PAR )

115 28 17 IDENT error

116 28 22 LEFT\_PAR (

117 28 23 IDENT err\_min\_value\_limit

118 28 42 RIGHT\_PAR )

119 28 43 SEMICOLON ;

120 29 13 ELSE\_SYMBOL else

121 30 17 IF\_SYMBOL if

122 30 19 LEFT\_PAR (

123 30 20 IDENT x

124 30 22 SUB -

125 30 24 IDENT y

126 30 25 RIGHT\_PAR )

127 31 21 RETURN\_SYMBOL return

128 31 28 LEFT\_PAR (

129 31 29 IDENT add

130 31 32 LEFT\_PAR (

131 31 33 IDENT x

132 31 34 COMMA ,

133 31 36 IDENT gauss\_sum

134 31 45 LEFT\_PAR (

135 31 46 IDENT x

136 31 48 PLUS +

137 31 50 INT 1

138 31 51 COMMA ,

139 31 53 IDENT y

140 31 54 RIGHT\_PAR )

141 31 55 RIGHT\_PAR )

142 31 56 RIGHT\_PAR )

143 31 57 SEMICOLON ;

144 32 17 ELSE\_SYMBOL else

145 33 21 RETURN\_SYMBOL return

146 33 28 LEFT\_PAR (

147 33 29 IDENT x

148 33 30 RIGHT\_PAR )

149 33 31 SEMICOLON ;

150 34 1 RIGHT\_BRACE }

151 36 1 VOID\_SYMBOL void

152 36 6 IDENT error

153 36 11 LEFT\_PAR (

154 36 12 INT\_SYMBOL int

155 36 16 IDENT err\_typ

156 36 23 RIGHT\_PAR )

157 36 24 LEFT\_BRACE {

158 37 5 SWITCH\_SYMBOL switch

159 37 11 LEFT\_PAR (

160 37 12 IDENT err\_typ

161 37 19 RIGHT\_PAR )

162 37 20 LEFT\_BRACE {

163 38 5 CASE\_SYMBOL case

164 38 10 INT 1

165 38 11 COLON :

166 39 9 PRINTF\_SYMBOL printf

167 39 15 LEFT\_PAR (

168 39 16 STRING Max value limit.

169 39 34 RIGHT\_PAR )

170 39 35 SEMICOLON ;

171 40 5 CASE\_SYMBOL case

172 40 10 INT 2

173 40 11 COLON :

174 41 9 PRINTF\_SYMBOL printf

175 41 15 LEFT\_PAR (

176 41 16 STRING Min value limit.

177 41 34 RIGHT\_PAR )

178 41 35 SEMICOLON ;

179 42 5 DEFAULT\_SYMBOL default

180 42 12 COLON :

181 43 9 PRINTF\_SYMBOL printf

182 43 15 LEFT\_PAR (

183 43 16 STRING Error Type Not Define.

184 43 40 RIGHT\_PAR )

185 43 41 SEMICOLON ;

186 44 5 RIGHT\_BRACE }

187 46 5 RETURN\_SYMBOL return

188 46 11 SEMICOLON ;

189 47 1 RIGHT\_BRACE }

190 49 1 VOID\_SYMBOL void

191 49 6 IDENT warning

192 49 13 LEFT\_BRACE {

193 50 5 PRINTF\_SYMBOL printf

194 50 11 LEFT\_PAR (

195 50 12 STRING The Program below is not clear.

196 50 45 RIGHT\_PAR )

197 50 46 SEMICOLON ;

198 51 5 RETURN\_SYMBOL return

199 51 11 SEMICOLON ;

200 52 1 RIGHT\_BRACE }

201 54 1 CHAR\_SYMBOL char

202 54 6 IDENT returnc

203 54 13 LEFT\_BRACE {

204 55 2 RETURN\_SYMBOL return

205 55 9 LEFT\_PAR (

206 55 10 CHAR C

207 55 13 RIGHT\_PAR )

208 55 14 SEMICOLON ;

209 56 1 RIGHT\_BRACE }

210 58 1 CHAR\_SYMBOL char

211 58 6 IDENT output

212 58 12 LEFT\_PAR (

213 58 13 CHAR\_SYMBOL char

214 58 18 IDENT input

215 58 23 RIGHT\_PAR )

216 58 24 LEFT\_BRACE {

217 59 2 PRINTF\_SYMBOL printf

218 59 8 LEFT\_PAR (

219 59 9 IDENT input

220 59 14 RIGHT\_PAR )

221 59 15 SEMICOLON ;

222 60 2 RETURN\_SYMBOL return

223 60 9 LEFT\_PAR (

224 60 10 IDENT input

225 60 15 RIGHT\_PAR )

226 60 16 SEMICOLON ;

227 61 1 RIGHT\_BRACE }

228 63 1 VOID\_SYMBOL void

229 63 6 IDENT test

230 63 10 LEFT\_BRACE {

231 64 5 INT\_SYMBOL int

232 64 9 IDENT int\_test

233 64 17 SEMICOLON ;

234 65 5 CHAR\_SYMBOL char

235 65 10 IDENT char\_test

236 65 19 SEMICOLON ;

237 66 5 INT\_SYMBOL int

238 66 9 IDENT \_abcdefghijklmnopqrstuvwxyz0123456789

239 66 46 SEMICOLON ;

240 67 5 CHAR\_SYMBOL char

241 67 10 IDENT letters

242 67 17 LEFT\_BRACKET [

243 67 18 INT 2

244 67 19 RIGHT\_BRACKET ]

245 67 20 SEMICOLON ;

246 69 5 IDENT int\_test

247 69 14 BECOME =

248 69 16 INT 1

249 69 17 MULT \*

250 69 18 INT 2

251 69 19 SEMICOLON ;

252 70 5 IDENT int\_test

253 70 14 BECOME =

254 70 16 INT 4

255 70 17 DIV /

256 70 18 INT 2

257 70 19 SEMICOLON ;

258 71 5 IDENT int\_test

259 71 14 BECOME =

260 71 16 INT 0

261 71 17 SEMICOLON ;

262 72 5 IDENT int\_test

263 72 14 BECOME =

264 72 16 PLUS +

265 72 17 INT 1234

266 72 21 SEMICOLON ;

267 73 5 IDENT int\_test

268 73 14 BECOME =

269 73 16 SUB -

270 73 17 INT 5678

271 73 21 SEMICOLON ;

272 74 5 IDENT char\_test

273 74 15 BECOME =

274 74 17 CHAR a

275 74 20 SEMICOLON ;

276 75 5 IDENT char\_test

277 75 15 BECOME =

278 75 17 CHAR A

279 75 20 SEMICOLON ;

280 76 5 IDENT letters

281 76 12 LEFT\_BRACKET [

282 76 13 INT 0

283 76 14 RIGHT\_BRACKET ]

284 76 16 BECOME =

285 76 18 CHAR 1

286 76 21 SEMICOLON ;

287 77 5 IDENT letters

288 77 12 LEFT\_BRACKET [

289 77 13 INT 1

290 77 14 RIGHT\_BRACKET ]

291 77 16 BECOME =

292 77 18 CHAR -

293 77 21 SEMICOLON ;

294 78 5 IDENT letters

295 78 12 LEFT\_BRACKET [

296 78 13 INT 2

297 78 14 RIGHT\_BRACKET ]

298 78 16 BECOME =

299 78 18 CHAR n

300 78 21 SEMICOLON ;

301 79 5 SEMICOLON ;

302 81 5 PRINTF\_SYMBOL printf

303 81 11 LEFT\_PAR (

304 81 12 IDENT letters

305 81 19 RIGHT\_PAR )

306 81 20 SEMICOLON ;

307 82 5 PRINTF\_SYMBOL printf

308 82 11 LEFT\_PAR (

309 82 12 IDENT letters

310 82 19 LEFT\_BRACKET [

311 82 20 INT 0

312 82 21 RIGHT\_BRACKET ]

313 82 22 RIGHT\_PAR )

314 82 23 SEMICOLON ;

315 83 1 RIGHT\_BRACE }

316 87 1 VOID\_SYMBOL void

317 87 6 MAIN\_SYMBOL main

318 87 10 LEFT\_PAR (

319 87 11 RIGHT\_PAR )

320 88 1 LEFT\_BRACE {

321 89 5 INT\_SYMBOL int

322 89 9 IDENT x

323 89 10 COMMA ,

324 89 11 IDENT y

325 89 12 SEMICOLON ;

326 90 5 CHAR\_SYMBOL char

327 90 10 IDENT input

328 90 15 SEMICOLON ;

329 91 5 CHAR\_SYMBOL char

330 91 10 IDENT c

331 91 11 SEMICOLON ;

332 92 5 INT\_SYMBOL int

333 92 9 IDENT result

334 92 15 LEFT\_BRACKET [

335 92 16 INT 100

336 92 19 RIGHT\_BRACKET ]

337 92 20 SEMICOLON ;

338 94 5 SCANF\_SYMBOL scanf

339 94 10 LEFT\_PAR (

340 94 11 IDENT x

341 94 12 COMMA ,

342 94 13 IDENT y

343 94 14 RIGHT\_PAR )

344 94 15 SEMICOLON ;

345 95 5 SCANF\_SYMBOL scanf

346 95 10 LEFT\_PAR (

347 95 11 IDENT input

348 95 16 RIGHT\_PAR )

349 95 17 SEMICOLON ;

350 97 5 IDENT i

351 97 7 BECOME =

352 97 9 INT 0

353 97 10 SEMICOLON ;

354 99 5 DO\_SYMBOL do

355 100 9 IDENT result

356 100 15 LEFT\_BRACKET [

357 100 16 IDENT i

358 100 17 RIGHT\_BRACKET ]

359 100 19 BECOME =

360 100 21 IDENT gauss\_sum

361 100 30 LEFT\_PAR (

362 100 31 IDENT x

363 100 32 COMMA ,

364 100 33 IDENT y

365 100 34 RIGHT\_PAR )

366 100 35 SEMICOLON ;

367 101 5 WHILE\_SYMBOL while

368 101 10 LEFT\_PAR (

369 101 11 INT 0

370 101 12 RIGHT\_PAR )

371 103 5 IDENT i

372 103 7 BECOME =

373 103 9 INT 0

374 103 10 SEMICOLON ;

375 104 5 DO\_SYMBOL do

376 104 7 LEFT\_BRACE {

377 105 9 IDENT result

378 105 15 LEFT\_BRACKET [

379 105 16 IDENT i

380 105 17 RIGHT\_BRACKET ]

381 105 19 BECOME =

382 105 21 IDENT gauss\_sum

383 105 30 LEFT\_PAR (

384 105 31 IDENT x

385 105 33 PLUS +

386 105 35 IDENT i

387 105 36 COMMA ,

388 105 38 IDENT y

389 105 39 RIGHT\_PAR )

390 105 40 SEMICOLON ;

391 106 9 IDENT i

392 106 11 BECOME =

393 106 13 IDENT i

394 106 15 PLUS +

395 106 17 INT 1

396 106 18 SEMICOLON ;

397 107 5 RIGHT\_BRACE }

398 107 6 WHILE\_SYMBOL while

399 107 11 LEFT\_PAR (

400 107 12 IDENT i

401 107 14 LSS <

402 107 16 IDENT y

403 107 18 SUB -

404 107 20 IDENT x

405 107 21 RIGHT\_PAR )

406 109 5 PRINTF\_SYMBOL printf

407 109 11 LEFT\_PAR (

408 109 12 STRING X =

409 109 18 COMMA ,

410 109 20 IDENT x

411 109 21 RIGHT\_PAR )

412 109 22 SEMICOLON ;

413 110 5 PRINTF\_SYMBOL printf

414 110 11 LEFT\_PAR (

415 110 12 STRING Y =

416 110 18 COMMA ,

417 110 20 IDENT y

418 110 21 RIGHT\_PAR )

419 110 22 SEMICOLON ;

420 111 5 PRINTF\_SYMBOL printf

421 111 11 LEFT\_PAR (

422 111 12 STRING X sum to Y =

423 111 27 RIGHT\_PAR )

424 111 28 SEMICOLON ;

425 112 5 PRINTF\_SYMBOL printf

426 112 11 LEFT\_PAR (

427 112 12 IDENT result

428 112 18 LEFT\_BRACKET [

429 112 19 INT 0

430 112 20 RIGHT\_BRACKET ]

431 112 21 RIGHT\_PAR )

432 112 22 SEMICOLON ;

433 113 5 PRINTF\_SYMBOL printf

434 113 11 LEFT\_PAR (

435 113 12 IDENT input

436 113 17 RIGHT\_PAR )

437 113 18 SEMICOLON ;

438 115 5 PRINTF\_SYMBOL printf

439 115 11 LEFT\_PAR (

440 115 12 STRING It's already

441 115 27 COMMA ,

442 115 29 IDENT returnyear

443 115 39 RIGHT\_PAR )

444 115 40 SEMICOLON ;

445 117 5 LEFT\_BRACE {

446 117 6 SEMICOLON ;

447 117 7 RIGHT\_BRACE }

448 118 5 LEFT\_BRACE {

449 118 6 SEMICOLON ;

450 118 7 LEFT\_BRACE {

451 118 8 SEMICOLON ;

452 118 9 RIGHT\_BRACE }

453 118 10 SEMICOLON ;

454 118 11 RIGHT\_BRACE }

455 121 5 IDENT c

456 121 7 BECOME =

457 121 9 IDENT output

458 121 15 LEFT\_PAR (

459 121 16 IDENT input

460 121 21 RIGHT\_PAR )

461 121 22 SEMICOLON ;

462 122 5 IDENT c

463 122 7 BECOME =

464 122 9 IDENT returnc

465 122 16 SEMICOLON ;

466 123 5 IDENT warning

467 123 12 SEMICOLON ;

468 124 5 IDENT test

469 124 9 SEMICOLON ;

470 125 1 RIGHT\_BRACE }